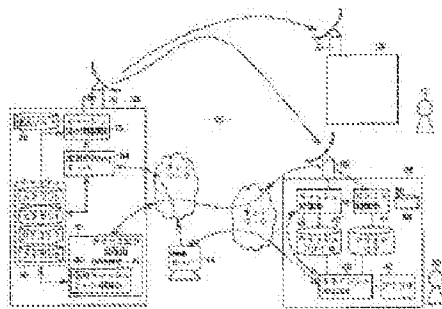


DISTRIBUTED FILE SYSTEM, BROADCASTING SERVER DEVICE FOR THE SYSTEM AND CLIENT DEVICE**Publication number:** JP11073361 (A)**Publication date:** 1999-03-16**Inventor(s):** DOI KATSUYOSHI; TODA HIROYOSHI; WAKABAYASHI HIROYO**Applicant(s):** SHARP KK**Classification:****- international:** **G06F12/00; G06F13/00; G06F12/00; G06F13/00; (IPC1-7): G06F12/00; G06F12/00; G06F13/00****- European:****Application number:** JP19970234917 19970829**Priority number(s):** JP19970234917 19970829**Abstract of JP 11073361 (A)**

PROBLEM TO BE SOLVED: To provide a distributed file system capable of accelerating responses of file objects in plural client computers.

SOLUTION: A broadcasting server device 22 includes a broadcasting proxy 64 for sorting file objects in each category and caches and broadcasting these sorted objects. A client device 26 includes a client proxy 46, a cache receiving part 48 for receiving file objects from the device 22 and caching only file objects belonging to a specific category and a category control part 50 for transmitting a file object list processed by the proxy 46 to the device 22. The device 22 calculates a virtual cache hit rate in each category based on a file object name list. The control part 50 sets up the category of file objects selected by the device 26 based on the virtual cache hit rate.



Data supplied from the **esp@cenet** database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-73361

(43) 公開日 平成11年(1999) 3月16日

(51) Int.Cl. ⁸	識別記号	F I	
G 0 6 F 12/00	5 4 5	G 0 6 F 12/00	5 4 5 Z
	5 3 3		5 3 3 J
13/00	3 5 7	13/00	3 5 7 Z

審査請求 未請求 請求項の数21 O L (全 34 頁)

(21) 出願番号 特願平9-234917

(22) 出願日 平成9年(1997) 8月29日

(71) 出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72) 発明者 土居 克良

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(72) 発明者 戸田 浩義

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(72) 発明者 若林 広世

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

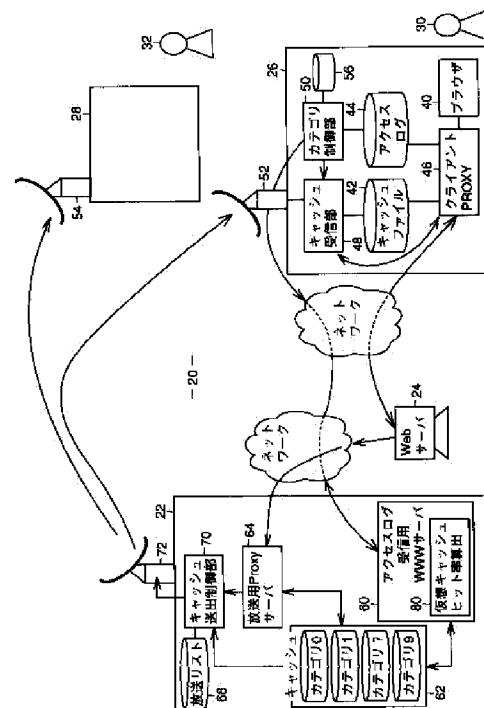
(74) 代理人 弁理士 深見 久郎

(54) 【発明の名称】 分散ファイルシステムならびに当該システムの放送サーバ装置およびクライアント装置

(57) 【要約】

【課題】 複数のクライアント計算機での、ファイルオブジェクトのレスポンスを高速化できる分散ファイルシステムを提供すること。

【解決手段】 放送サーバ装置22は、ファイルオブジェクトをカテゴリ別に分類してキャッシュし放送する放送用プロキシ64を含む。クライアント装置26は、クライアントプロキシ46と、放送サーバ装置22からファイルオブジェクトを受信し、特定のカテゴリのものをキャッシュするキャッシュ受信部48と、クライアントプロキシが処理したファイルオブジェクトリストを放送サーバ装置22に送信するカテゴリ制御部50とを含む。放送サーバ装置22は、ファイルオブジェクト名称リストに基づき、カテゴリ別の仮想的なキャッシュヒット率を計算する。カテゴリ制御部50は、仮想的なキャッシュヒット率に基づいて、クライアント装置が選別するファイルオブジェクトのカテゴリを設定する。



【特許請求の範囲】

【請求項1】 ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および前記放送サーバ装置と通信可能で、前記放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムであって、

前記放送サーバ装置は、ファイルオブジェクトを複数個のカテゴリに分類してキャッシュし、カテゴリ識別情報とともに放送するための放送用プロキシサーバ手段を含み、

前記クライアント装置は、

キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するためのプロキシサーバ手段と、

前記放送サーバ装置によって放送されたファイルオブジェクトを受信し、予め設定されたカテゴリのファイルオブジェクトのみを選択して前記プロキシサーバ手段に与えてキャッシュさせるための選別手段と、

前記プロキシサーバ手段に記録されたファイルオブジェクト名称のリストを前記放送サーバ装置に送信するカテゴリ制御手段とを含み、

前記放送サーバ装置はさらに、前記リスト送信手段により送信されたファイルオブジェクト名称のリストに基づき、各カテゴリ別の仮想的なキャッシュヒット率を計算して前記クライアント装置に提示するためのキャッシュヒット率計算手段を含み、

前記カテゴリ制御部は、前記キャッシュヒット率計算手段により提示された仮想的なキャッシュヒット率に基づいて、前記クライアント装置におけるキャッシュヒット率が高くなるように、前記選別手段により選別されるファイルオブジェクトのカテゴリを設定する、分散ファイルシステム。

【請求項2】 前記放送サーバ装置は、前記各カテゴリ別のキャッシュサイズを前記クライアント装置に通知するための手段をさらに含み、

前記カテゴリ制御手段は、前記放送サーバ装置から通知された前記各カテゴリ別のキャッシュサイズと、前記プロキシサーバ装置のキャッシュの割当領域とから、前記選別手段により選別可能なカテゴリ数を定め、前記選別手段の選別するカテゴリ数の上限として設定する、請求項1に記載の分散ファイルシステム。

【請求項3】 前記クライアント装置は、

特定のカテゴリ番号を記憶する手段をさらに含み、

前記選別手段は、前記特定のカテゴリ番号のファイルオブジェクトを排除して前記プロキシサーバ手段に与える、請求項1または2に記載の分散ファイルシステム。

【請求項4】 前記クライアント装置は、

特定のカテゴリ番号を記憶する手段をさらに含み、

前記選別手段は、前記特定のカテゴリ番号のファイルオ

ブジェクトを常に選択して前記プロキシサーバ手段に与える、請求項1または2に記載の分散ファイルシステム。

【請求項5】 前記カテゴリ制御手段は、前記プロキシサーバ手段に記録されたファイルオブジェクト名称のうち、テキストデータに相当するファイルオブジェクト名称のみを選択して前記リストとして送信する、請求項1に記載の分散ファイルシステム。

【請求項6】 ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および前記放送サーバ装置と通信可能なクライアント装置とを含むファイル配送のための分散ファイルシステムであって、

前記放送サーバ装置は、サーバ装置から受けたファイルオブジェクトを複数個のカテゴリに分類してキャッシュし、カテゴリ識別情報とともに放送するための放送プロキシサーバ手段を含み、

前記クライアント装置は、

キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するとともに、アクセスされたファイルオブジェクト名称を記録するためのプロキシサーバ手段と、

前記放送サーバ装置により放送されるファイルオブジェクトを受信して前記プロキシサーバ手段にキャッシュさせるための手段と、

前記プロキシサーバ手段により記録されたファイルオブジェクト名称のリストを前記放送サーバ装置に送信するための手段とを含み、

前記放送サーバ装置はさらに、送信された前記ファイルオブジェクト名称のリストに基づいてサーバ装置からファイルオブジェクトを収集して前記放送手段に与え、クライアント装置に向けて放送させるための手段を含む、分散ファイルシステム。

【請求項7】 ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および前記放送サーバ装置と通信可能で、前記放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムであって、

前記放送サーバ装置は、

キャッシュを有し、サーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するための放送用プロキシサーバ手段と、

所定の方式にしたがって定められる時刻に、前記キャッシュ中のファイルオブジェクトの、サーバ装置における最終変更時刻情報を放送するための放送手段とを含み、

前記クライアント装置は、

キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理し、アクセスされたファイルオブジェクトの最終変更時刻を記録するためのプロキシサーバ手段と、

ユーザによるファイルオブジェクトへのアクセス要求があったときに、前記プロキシサーバ手段の記録を参照して、当該ファイルオブジェクトの最終ファイル変更時刻と予め定められたキャッシュ有効時間とに基づいて、サーバ装置へのアクセス要求を発行する処理と、前記キャッシュから当該ファイルオブジェクトを読みだしてユーザに返す処理とのいずれかを選択的に実行するための手段と、

前記放送サーバ装置から受信した最終変更時刻情報に基づき、前記プロキシサーバ手段により記録された各ファイルオブジェクトの最終ファイル変更時刻を更新する処理と、無効と判定されたファイルオブジェクトに対するアクセス要求をプロキシサーバ手段に対して発行する処理とを選択的に実行するための手段とを含む、分散ファイルシステム。

【請求項8】 放送サーバ装置と、前記放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムであって、

前記放送サーバ装置は、ファイルオブジェクトを複数個のカテゴリ別にキャッシュし、カテゴリ識別情報とともに放送するための手段を含み、

前記クライアント装置は、

キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するとともに、アクセスされたファイルオブジェクト名称を記録するためのプロキシサーバ手段と、

前記放送サーバ装置によって放送されたファイルオブジェクトを受信し、予め設定されたカテゴリ識別情報を有するファイルオブジェクトのみを選別して前記プロキシサーバ手段に与えカテゴリ識別情報とともにキャッシュさせるための選別手段と、

前記プロキシサーバ手段によって処理された最新のカテゴリ識別情報を記憶するための手段と、

前記プロキシサーバ手段に記録されたファイルオブジェクト名称のカテゴリ別リストを、前記放送サーバ装置に送信するためのカテゴリ別リスト送信手段と、

前記プロキシサーバ手段によるサーバ装置へのアクセスがあったときに、アクセスされたファイルオブジェクトのカテゴリ識別情報として前記記憶された最新のカテゴリ識別情報を設定するための手段とを含み、

前記放送サーバ装置はさらに、前記カテゴリ別リスト送信手段により送信されたファイルオブジェクト名称のカテゴリ別リストに基づいてサーバ装置からファイルオブジェクトを収集して前記放送手段に与え、クライアント装置に向けて放送させてクライアント装置の前記キャッシュにカテゴリ別に蓄積させるための手段を含む、分散ファイルシステム。

【請求項9】 ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および前記放送サーバ

装置と通信可能で、前記放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムにおいて用いられる放送サーバ装置であって、

ファイルオブジェクトを複数個のカテゴリに分類してキャッシュし、カテゴリ識別情報とともに放送するための放送用プロキシサーバ手段と、

クライアント装置から送信されてくる、前記クライアント装置によってアクセスされたファイルオブジェクト名称のリストに基づき、前記クライアント装置における各カテゴリ別の仮想的なキャッシュヒット率を計算して前記クライアント装置に提示するためのキャッシュヒット率計算手段を含む、放送サーバ装置。

【請求項10】 前記放送サーバ装置は、前記各カテゴリ別のキャッシュサイズを前記クライアント装置に通知するための手段をさらに含む、請求項9に記載の放送サーバ装置。

【請求項11】 ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および前記放送サーバ装置と通信可能なクライアント装置とを含むファイル配送のための分散ファイルシステムにおいて用いられる放送サーバ装置であって、

サーバ装置から受けたファイルオブジェクトを複数個のカテゴリに分類してキャッシュし、カテゴリ識別情報とともに放送するための放送プロキシサーバ手段と、

クライアント装置から送信された、前記クライアント装置がアクセスしたファイルオブジェクト名称のリストに基づいてサーバ装置からファイルオブジェクトを収集して前記放送手段に与え、クライアント装置に向けて放送させるための手段とを含む、放送サーバ装置。

【請求項12】 ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および前記放送サーバ装置と通信可能で、前記放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムにおいて用いられる放送サーバ装置であって、

キャッシュを有し、サーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するための放送用プロキシサーバ手段と、

所定の方式にしたがって定められる時刻に、前記キャッシュ中のファイルオブジェクトの、サーバ装置における最終変更時刻情報を放送するための放送手段とを含む、放送サーバ装置。

【請求項13】 放送サーバ装置と、前記放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムにおいて用いられる放送サーバ装置であって、

ファイルオブジェクトを複数個のカテゴリ別にキャッシュし、カテゴリ識別情報とともに放送するための手段

と、クライアント装置から送信された、前記クライアント装置によってアクセスされたファイルオブジェクト名称のカテゴリ別リストに基づいてサーバ装置からファイルオブジェクトを収集して前記放送手段に与え、クライアント装置に向けて放送させてクライアント装置の前記キャッシュにカテゴリ別に蓄積させるための手段を含む、放送サーバ装置。

【請求項14】 ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および前記放送サーバ装置と通信可能で、前記放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムにおいて用いられるクライアント装置であって、前記放送サーバ装置は、ファイルオブジェクトを複数個のカテゴリに分類してキャッシュし、カテゴリ識別情報とともに放送するための放送用プロキシサーバ手段を含み、前記クライアント装置は、キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するためのプロキシサーバ手段と、前記放送サーバ装置によって放送されたファイルオブジェクトを受信し、予め設定されたカテゴリのファイルオブジェクトのみを選別して前記プロキシサーバ手段に与えてキャッシュさせるための選別手段と、前記プロキシサーバ手段に記録されたファイルオブジェクト名称のリストを前記放送サーバ装置に送信するためのカテゴリ制御手段とを含み、前記放送サーバ装置はさらに、前記カテゴリ制御手段により送信されたファイルオブジェクト名称のリストに基づき、各カテゴリ別の仮想的なキャッシュヒット率を計算して前記クライアント装置に提示するためのキャッシュヒット率計算手段を含み、前記カテゴリ制御手段は、前記キャッシュヒット率計算手段により提示された仮想的なキャッシュヒット率に基づいて、前記クライアント装置におけるキャッシュヒット率が高くなるように、前記選別手段により選別されるファイルオブジェクトのカテゴリを設定する、クライアント装置。

【請求項15】 前記放送サーバ装置は、前記各カテゴリ別のキャッシュサイズを前記クライアント装置に通知するための手段をさらに含み、前記カテゴリ制御手段は、前記放送サーバ装置から通知された前記各カテゴリ別のキャッシュサイズと、前記プロキシサーバ装置のキャッシュの割当領域とから、前記選別手段により選別可能なカテゴリ数を定め、前記選別手段の選別するカテゴリ数の上限として設定する、請求項14に記載のクライアント装置。

【請求項16】 前記クライアント装置は、

特定のカテゴリ番号を記憶する手段をさらに含み、前記選別手段は、前記特定のカテゴリ番号のファイルオブジェクトを排除して前記プロキシサーバ手段に与える、請求項14または15に記載のクライアント装置

【請求項17】 前記クライアント装置は、特定のカテゴリ番号を記憶する手段をさらに含み、前記選別手段は、前記特定のカテゴリ番号のファイルオブジェクトを常に関連して前記プロキシサーバ手段に与える、請求項14または15に記載のクライアント装置。

【請求項18】 前記カテゴリ制御手段は、前記プロキシサーバ手段に記録されたファイルオブジェクト名称のうち、テキストデータに相当するファイルオブジェクト名称のみを選択してファイルオブジェクト名称のリストとして前記放送サーバ装置に与える、請求項14に記載のクライアント装置。

【請求項19】 ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および前記放送サーバ装置と通信可能なクライアント装置とを含むファイル配送のための分散ファイルシステムにおいて用いられるクライアント装置であって、

前記放送サーバ装置は、サーバ装置から受けたファイルオブジェクトを複数個のカテゴリに分類してキャッシュし、カテゴリ識別情報とともに放送するための放送プロキシサーバ手段を含み、

前記クライアント装置は、キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するとともに、アクセスされたファイルオブジェクト名称を記録するためのプロキシサーバ手段と、前記放送サーバ装置により放送されるファイルオブジェクトを受信して前記プロキシサーバ手段にキャッシュさせるための手段と、

前記プロキシサーバ手段により記録されたファイルオブジェクト名称のリストを前記放送サーバ装置に送信するための手段とを含む、クライアント装置。

【請求項20】 ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および前記放送サーバ装置と通信可能で、前記放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムにおいて用いられるクライアント装置であって、

前記放送サーバ装置は、キャッシュを有し、サーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するための放送用プロキシサーバ手段と、

所定の方式にしたがって定められる時刻に、前記キャッシュされたファイルオブジェクトの、サーバ装置における最終変更時刻情報を放送するための放送手段とを含み、

前記クライアント装置は、

キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するとともに、アクセスされたファイルオブジェクト名称を最終変更時刻とともに記録するためのプロキシサーバ手段と、

ユーザによるファイルオブジェクトへのアクセス要求があったときに、前記プロキシサーバ手段の記録を参照して、当該ファイルオブジェクトの最終ファイル変更時刻と予め定められたキャッシュ有効時間とに基づいて、サーバ装置へのアクセス要求を前記放送サーバ装置に発行する処理と、前記キャッシュから当該ファイルオブジェクトを読みだしてユーザに返す処理とのいずれかを選択するための手段と、

前記放送サーバ装置から受信した最終変更時刻情報に基づき、前記プロキシサーバ手段により記録された各ファイルオブジェクトの最終ファイル変更時刻を更新する処理と、無効と判定されたファイルオブジェクトに対するアクセス要求をプロキシサーバ手段に対して発行する処理とを選択的に実行するための手段とを含む、クライアント装置。

【請求項21】 放送サーバ装置と、前記放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムにおいて用いられるクライアント装置であって、

前記放送サーバ装置は、ファイルオブジェクトを複数個のカテゴリ別にキャッシュし、カテゴリ識別情報とともに放送するための手段を含み、

前記クライアント装置は、

キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するとともに、アクセスされたファイルオブジェクト名称を記録するためのプロキシサーバ手段と、

前記放送サーバ装置によって放送されたファイルオブジェクトを受信し、予め設定されたカテゴリ識別情報を有するファイルオブジェクトのみを選別して前記プロキシサーバ手段にカテゴリ識別情報とともに与えてキャッシュさせるための選別手段と、

前記プロキシサーバ手段によって処理された最新のカテゴリ識別情報を記憶するための手段と、

前記プロキシサーバ手段に記録されたファイルオブジェクト名称のカテゴリ別リストを、前記放送サーバ装置に送信するための手段と、

前記プロキシサーバ手段によるサーバ装置へのアクセスがあったときに、アクセスされたファイルオブジェクトのカテゴリ識別情報として前記記憶された最新のカテゴリ識別情報を設定するためのカテゴリ手段とを含む、クライアント装置。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】本発明は、ネットワーク上に分散した複数のサーバ装置（サーバ計算機）と、複数のクライアント装置（クライアント計算機）と、サーバ計算機のファイルオブジェクトをキャッシュしてクライアント計算機に対して放送するキャッシュ放送サーバ装置（キャッシュ放送サーバ計算機）とが通信回線で相互接続されている分散ファイルシステムに関し、特に、トラフィックの増大を回避しながら、クライアント計算機におけるキャッシュヒット率を向上させるための技術に関する。

【0002】

【従来の技術】以下の記載において、「放送」とは、有線LAN（ローカルエリアネットワーク）におけるIP Multicastを利用する放送、デジタル衛星放送、または地上波テレビジョン（TV）放送の垂直帰線期間にデジタルデータを挿入するVBI（Vertical Blanking Interval）等のデータ放送等のデータ放送インフラストラクチャーを利用するものをいう。また、クライアント計算機としては、パーソナルコンピュータまたは専用の受信セットトップボックス等を想定する。サーバ計算機は、World Wide Web(WWW)等の情報提供システムのサーバシステムに相当する。

【0003】従来の分散ファイルシステムにおいては、複数のクライアント計算機からのサーバ計算機上のファイルオブジェクトに対する読出要求（アクセス要求）は、一旦途中のゲートウェイ計算機で中継されていた。ここで、ファイルオブジェクトとは、ファイルシステムの利用するネットワークプロトコルと、ネットワークアドレス（サーバ計算機の名称）と、ファイル名称と、ファイルの実体との組を言う。また、「ファイルオブジェクトの名称（ファイルオブジェクト名称）」とは、ネットワークプロトコルと、ネットワークアドレスと、ファイル名称との組のことを言う。

【0004】ゲートウェイ計算機は、たとえば特開平4-313126号公報に開示されているように、ディスク装置、半導体メモリ等からなるキャッシュファイルを備えており、クライアント計算機からアクセス要求を受けると、アクセス要求のあったファイルオブジェクトがキャッシュされているか否かを調べる。アクセス要求のあったファイルオブジェクトがキャッシュされていない場合には、ゲートウェイ計算機はサーバ計算機から当該ファイルオブジェクトを読出し、アクセス要求を出したクライアント計算機に対して中継転送する。またこのときゲートウェイ計算機は、このファイルオブジェクトをキャッシュファイルに書き込む。こうしてキャッシュファイルにはファイルオブジェクトが蓄積されて行く。

【0005】アクセス要求のあったファイルオブジェクトがキャッシュされている場合、ゲートウェイ計算機はまず、キャッシュされているファイルオブジェクトと、

サーバ計算機の当該ファイルオブジェクトとの最終変更時間を比較する。キャッシュの内容が古くサーバ計算機の内容が新しければ、ゲートウェイ計算機はサーバ装置から当該ファイルオブジェクトを読み出し、アクセス要求を出したクライアント計算機にこのファイルオブジェクトを中継するとともに、ゲートウェイ計算機内のキャッシュ内のファイルオブジェクトを、新しく読み出されたファイルオブジェクトで更新する。

【0006】キャッシュの内容が有効であれば、サーバ計算機からファイルオブジェクトを改めて読み出すことなく、キャッシュの内容を読み出してアクセス要求を出したクライアント計算機に対して送信する。

【0007】従来はキャッシュ付ゲートウェイ計算機はこのようなしてクライアント計算機からのアクセス要求を処理していた。

【0008】いわゆるインターネットにおけるこうしたキャッシュ付ゲートウェイ計算機はProxyサーバと呼ばれる。インターネットとは、その基本として、TCP/IPプロトコルを利用したグローバルなネットワークである。インターネットにおけるTCP/IPを利用した地域分散型マルチメディア情報提供システムが前述のWWWシステムである。

【0009】WWWシステムは、ネットワーク上に分散したファイルオブジェクトを扱うことができる。これらのファイルオブジェクトは、テキスト、画像、音声、ビデオ画像等の多様な種類のデータを含み、情報提供者側にとっても、情報利用者（ユーザ）にとっても魅力的であるため、ネットワーク上におけるWWWのトラフィックが爆発的に増加しつつある。

【0010】WWWシステムの普及の要因として、ユーザ側における操作の簡便さがある。クライアント計算機のユーザは、グラフィカルユーザインターフェイスをもったブラウザソフトウェア（以下単に「ブラウザ」と呼ぶ。）をクライアント計算機に搭載し、動作させるだけで、世界的規模のネットワーク上に分散したサーバ計算機の保持するファイルオブジェクトで構成された情報を次々とアクセスすることができる。ファイルオブジェクトの存在場所は、ブラウザ上に表示される画面に埋め込まれているため、ファイルオブジェクトの存在場所が予め分からなくとも簡単な操作を行なうだけでそのファイルオブジェクトにアクセスすることができる。こうした操作の簡単さが、WWWシステムの普及の大きな要因である。

【0011】WWWシステムでは、TCP/IPプロトコル上に構築されたHypertext Transfer Protocol(HTTP)でファイルオブジェクトの転送を行なっている。そしてHTTPにしたがったファイル転送では、前述のゲートウェイ計算機によりファイルオブジェクトをキャッシュし中継転送する方式が広く行なわれている。インターネットにおけるこうしたProxyサーバは、クライアント計算機からネ

ットワーク上のWWWサーバ計算機のファイルオブジェクトに対するアクセス要求があった場合に、そのアクセス要求をサーバ計算機を代理して受け付けることにより、ネットワーク上のトラフィックを低減させ、アクセス要求に対するレスポンスを高速化させる。

【0012】Proxyサーバの概念を図12に示す。図12を参照して、分散ファイルシステム（インターネット）280は、外部ネットワーク300と内部ネットワーク302との間を接続するProxyサーバであるゲートウェイ計算機290と、外部ネットワーク300上の多数のサーバ計算機292を含む。内部ネットワーク302には、多数のクライアント計算機294、296等が接続されており、それぞれユーザ304等がブラウザ40を用いてサーバ計算機292のファイルオブジェクト298をアクセスする。

【0013】Proxyサーバであるゲートウェイ計算機（以下「Proxyサーバ」と呼ぶ。）290は、内部で動作するProxyプロセス320と、Proxyプロセス320の利用するキャッシュファイル324と、Proxyプロセス320が、キャッシュされたファイルに対するアクセス要求を出したクライアント計算機名称、その時刻、アクセス要求が出されたファイルオブジェクト名称とを記録するためのアクセスログ322を含む。

【0014】図13を参照して、Proxyサーバ290は、物理的にはネットワークインターフェースを持った計算機であって、CPU（中央演算処理装置）340と、CPU340が接続された内部バス342と、内部バス342と内部ネットワーク（LAN）302とに接続されるネットワークI/O部346と、I/O部348を介して内部バス342に接続されたファイル装置350を含む。図13に示す例では、物理的には内部ネットワーク302が外部ネットワーク300に接続されているが、論理的には内部ネットワーク302のクライアント計算機294は、必ずProxyサーバ290を介して外部ネットワーク300に接続される。キャッシュファイル、アクセスログ、およびProxyプロセス320の用いる変数等は、ファイル装置350およびメモリ344からなるファイルシステム内に確保される。またProxyプロセス自体は、CPU340により実行されるソフトウェアによって実現される。Proxyプロセスを実現するソフトウェアとしては、W3C HTTPD、DeleGate等と呼ばれるものがある。

【0015】再び図12を参照して、Proxyサーバ290は以下のように動作する。ユーザ304が、クライアント計算機294上でブラウザ40を動作させ、サーバ計算機292のファイルオブジェクト298を取得するよう、Proxyサーバ290に対して内部ネットワーク302を経由してアクセス要求を出す。

【0016】Proxyサーバ290内のProxyプロセス320がこのアクセス要求を受ける。Proxyプロセス32

0は、このアクセス要求を受けると、キャッシュファイル324をアクセスし、キャッシュされているデータの有無を調べる。当該ファイルオブジェクトがキャッシュされていれば、Proxy プロセス320は、Proxy プロセス320固有のファイル有効期限と、キャッシュされたファイルオブジェクトの最終ファイル変更時刻（タイムスタンプ）とを比較して、キャッシュされたファイルオブジェクトが有効かどうかを判定する。キャッシュされたファイルオブジェクトが有効であれば、キャッシュファイル324から当該ファイルオブジェクトを読み出し、内部ネットワーク302を介して、アクセス要求を出したクライアント計算機294にデータを送信する。

【0017】キャッシュファイル324に有効なファイルオブジェクトがなければ、すなわちキャッシュされているファイルオブジェクトの有効期限が過ぎているか、またはキャッシュファイル324に当該ファイルオブジェクトがキャッシュされていなければ、Proxy プロセス320は、外部ネットワーク300を介してサーバ計算機292に対しこのファイルオブジェクトのアクセス要求を出す。

【0018】サーバ計算機292は、このアクセス要求を受けると、指定されたファイルオブジェクト298を読み出し、外部ネットワーク300を経由してProxy プロセス320に送信する。Proxy プロセス320は、このファイルオブジェクトを受信すると、ログファイル322に、アクセスログを書き出す。Proxy プロセス320は同時に、このファイルオブジェクトを内部ネットワーク302を介して、アクセス要求を出したクライアント計算機294にデータを送信し、さらにキャッシュファイル324にキャッシュする。このとき、転送時のHTTPヘッダ情報、ファイルオブジェクトの書込み日付、アクセス日時がキャッシュファイル324に記録される。

$$T_0 = (1-H) \times T_x + H \times T_i$$

ここで、 T_x は外部ネットワークを利用したあるファイルオブジェクトの転送時間を、 T_i はProxy サーバとクライアント計算機との間の内部ネットワークを介した当該ファイルオブジェクトの転送時間を、それぞれ示す。

【0023】通常、内部ネットワークは十分高速であり

$$T_0 = (1-H) \times T_x$$

キャッシュなしの場合の転送時間は T_x であるから、キャッシュを使用することにより転送時間が $(1-H)$ 倍に短縮される。たとえばキャッシュヒット率 $H=0.5$ （50%）であれば、時間は $1-0.5=0.5$ 、すなわち50%に短縮される。キャッシュヒット率 $H=0.3$ （30%）であれば時間は $1-0.3=0.7$ 、すなわち70%に短縮される。すなわち、待ち時間が30%に短縮される。これが、複数のクライアント計算機によって単一のキャッシュファイルを共有することによる効果である。

【0025】キャッシュ蓄積量が増大するとヒットする

【0019】別のクライアント計算機296のブラウザ40から同じファイルオブジェクトに対するアクセス要求をProxy プロセス320に出したときにも、前述と同様の処理がProxy プロセス320によって行なわれる。このアクセス時が、キャッシュファイル324にキャッシュされた当該ファイルオブジェクトの有効期限内であれば、サーバ計算機292ではなくキャッシュファイル324から当該ファイルオブジェクトが読み出されてクライアント計算機296に転送される。つまり、複数のクライアント計算機が同一のProxy サーバ290のキャッシュファイル324を共有するので、多くのクライアントからのアクセスがあるファイルオブジェクトはキャッシュファイル324から取り出せる確率が高くなる。

【0020】図14に示すように、クライアント計算機360内にProxy サーバを構成するProxy プロセス46を内蔵することも可能である。この場合、図14を参照して、キャッシュファイル42およびアクセスログ44もクライアント計算機360内で準備される。なお、図14において、図12と同一の部品（構成要素）には同一の参照符号を付してある。それらの機能も同一であるので、それらについての詳しい説明は繰返さない。なお、Proxy プロセス46、キャッシュファイル42、およびアクセスログ44はそれぞれ図12に示すProxy プロセス320、キャッシュファイル324、およびアクセスログ322と同一のものである。

【0021】ゲートウェイ計算機（Proxy サーバ）の性能の指標の一つにキャッシュヒット率がある。キャッシュヒット率を H とすると、ファイルオブジェクトの転送時間 T_0 は以下のようにして計算できる。

【0022】

【数1】

$$\dots (1)$$

T_i は0とみなすことができる。したがってこの場合、式(1)は次の式(2)で近似できる。

【0024】

【数2】

$$\dots (2)$$

確率が高くなる。複数のクライアント計算機が同一キャッシュを利用するため、あるクライアント計算機がアクセスするファイルオブジェクトは既に他のクライアント計算機によってアクセスされている場合が多くなるからである。

【0026】キャッシュヒット率を向上させる方策として、以下が知られている。まず、できるだけ興味の似た複数のクライアント計算機ユーザによって同一キャッシュを共有させるようにすること、および、できるだけ大容量のファイルシステムに多数のファイルオブジェクトをキャッシュすること、である。こうすることで、再度

同一のファイルオブジェクトが利用される確率が高まるからである。

【0027】キャッシュヒット率を向上させるための技術として、クライアント計算機が利用するProxy サーバのアクセスログをもとに、そのProxy サーバを利用する他の複数のクライアント計算機にキャッシュファイルのファイルオブジェクトを放送する方式が考えられる。この方式によって、多数のクライアント計算機が、単一のキャッシュファイルを仮想的に共有する。しかもこの場合、キャッシュファイルは各クライアント計算機内部に存在することになる。あるクライアント計算機がアクセスしたファイルオブジェクトが他のクライアント計算機のキャッシュファイルにも配送され蓄積されるので、各クライアント計算機のキャッシュ蓄積量が増大し、キャッシュヒット率が向上することが期待され、ひいてはアクセス速度が改善されることが期待される。

【0028】発明者の経験によれば、ある企業の数百人のユーザ集団でキャッシュファイルを共有した場合に、キャッシュ蓄積量0MB（メガバイト）ではヒット率0%、100MB程度で30%、300MB程度で40%、そして1GB（ギガバイト）では50%位となる。すなわちキャッシュ蓄積量の増加に対応してヒット率も向上する。ただし、キャッシュ蓄積量の増加に対するキャッシュヒット率の伸びは50%程度までで鈍化する傾向がある。また、ユーザ数が増大し、ユーザの興味を持つサーバ計算機の対象が広がるとキャッシュヒット率は低下する傾向がある。たとえばいわゆるインターネットプロバイダ等、不特定多数のユーザが対象のゲートウェイ計算機では、20GB程度のキャッシュファイル蓄積量でもヒット率は40%程度である。

【0029】ところで、WWWの世界では、ファイルオブジェクトの変化率は1日あたり5%程度であることが観測されている。したがってほとんどのファイルオブジェクトは変化しないということが出来る。しかし、キャッシュ有効期限を24時間以内程度に設定しておかなければ、既に古くなった内容のファイルオブジェクトがキャッシュされておりユーザがそれを読出してしまふことが有り得る。

【0030】そこで、ユーザのアクセスログをもとに、ユーザのアクセス要求に対する処理とは別に、周期的にProxy サーバのキャッシュファイルの自動更新を行なうことが考えられる。たとえば夜間等、ネットワークトラフィックの少ない時間帯に自動でサーバ装置からファイルオブジェクトを取得してキャッシュ更新処理を行なう。この周期をキャッシュ有効期限と同程度とすれば、キャッシュ内のこれらファイルオブジェクトについては、常に有効期限以内であることが保証できる。

【0031】具体的には、前回の自動更新後、キャッシュ有効期限切れになる前に、キャッシュされているファイルオブジェクトについてサーバ計算機に対してその最

新性のチェックをかける。更新されていればそのファイルオブジェクトをサーバ計算機から取得してキャッシュすることによりキャッシュされているデータを最新データとする。変化がなければキャッシュファイルシステムの最終変更時刻をその時刻に更新し、キャッシュの最新性を確保する。

【0032】これによりユーザが利用する時間帯にはキャッシュの最新性チェックを省略し、ただちにキャッシュファイルからオブジェクトを取り出せるメリットがあり、応答時間の短縮に効果がある。

【0033】

【発明が解決しようとする課題】しかし、これと同様の処理をクライアントProxy キャッシュにおいて行えばクライアント計算機の数だけキャッシュ更新アクセスがサーバ計算機に集中する。したがってトラフィックが爆発的に増加するという問題がある。また、こうした処理には長時間にわたってキャッシュ更新処理のためネットワーク接続を行う必要があり、専用線で接続されたProxy サーバ計算機以外では実現が困難であるという問題もある。

【0034】さらに、以下の様な課題もある。前述のキャッシュ放送方式では、デジタル衛星放送を利用すれば1Mbps以上の伝送速度は簡単に確保でき、24時間放送すれば10GB程度の放送が可能である。しかしクライアント計算機ではパーソナルコンピュータのように記憶媒体の大きさは1GB程度であるから10GBもの蓄積は不可能である。したがって、受信されたファイルオブジェクトを何らかの方式にしたがって取捨選択してキャッシュすることが必要になる。したがって、クライアント計算機のユーザが必要としているものだけを蓄積し、残りは捨てる機構を提供することが課題となる。

【0035】また他の課題として次のようなものがある。従来のキャッシュ放送方式では、放送するためのキャッシュ付きProxy サーバ計算機が用意されていた。そして、クライアント計算機がこのProxy サーバ計算機を利用してサーバ計算機にアクセスすることで、どのようなファイルオブジェクトが利用されるかを調査し、どのファイルオブジェクトをキャッシュして放送すべきかを決定していた。

【0036】しかしこの場合、クライアント計算機が多数であればProxy サーバ計算機へのアクセスが集中し処理負荷が過大となるという問題となる。Proxy サーバ計算機であるゲートウェイ計算機のファイルオブジェクト中継能力は有限であり、利用できるクライアント計算機の台数には限度がある。例えばCERN httpd proxyでは毎秒5から10ファイルオブジェクトの中継しかできない。これはクライアント計算機台数にして百台～数百台程度までからのアクセスの処理能力にしか相当しないことが経験上知られている。

【0037】また、クライアント計算機とゲートウェイ

計算機とが遠隔地にありネットワークが細い場合もありうる。すなわちクライアント計算機は地方Aのインターネットプロバイダに接続されたパーソナルコンピュータであるが、キャッシュを共有し放送するためのProxyサーバ計算機は衛星放送基地Bにありその間のネットワークが必ずしも近くて帯域が十分あるとは言えない場合がありうる。このようなボトルネックが生ずるのは、クライアント計算機同士がキャッシュファイルオブジェクトを共有するためにゲートウェイ計算機のキャッシュを利用するというアーキテクチャを採用したことが原因である。

【0038】さらに、次のような課題がある。従来の単体のキャッシュ能動的更新型Proxyサーバ技術ではキャッシュ更新作業の結果が該当Proxyサーバにのみ利用される。他のProxyサーバにはそうしたキャッシュ更新作業の結果が反映されない。例えば前述のクライアント計算機の内蔵するProxyサーバが能動的にキャッシュを更新する場合その成果は他のクライアントとは共有できない。したがって、ネットワークおよびサーバ計算機に対するキャッシュ更新の負荷はクライアントの数だけ増加することになる。

【0039】この発明は、上述した問題を解決するためになされたもので、請求項1～8に記載の発明の目的は、複数のクライアント計算機での、ファイルオブジェクトのレスポンスを高速化できる分散ファイルシステムを提供することである。

【0040】請求項9～13に記載の発明の目的は、複数のクライアント計算機での、ファイルオブジェクトのレスポンスを高速化できる放送サーバ装置を提供することである。

【0041】請求項14～21に記載の発明の目的は、複数のクライアント計算機が接続される分散ファイルシステム内で用いられる、ファイルオブジェクトのレスポンスを高速化できるクライアント装置を提供することである。

【0042】

【課題を解決するための手段】請求項1に記載の分散ファイルシステムはネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および放送サーバ装置と通信可能で、放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムである。放送サーバ装置は、ファイルオブジェクトを複数個のカテゴリに分類してキャッシュし、カテゴリ識別情報とともに放送するための放送用プロキシサーバ手段を含む。クライアント装置は、キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するためのプロキシサーバ手段と、放送サーバ装置によって放送されたファイルオブジェクトを受信し、予め設定されたカテゴリのファイ

ルオブジェクトのみを選別してプロキシサーバ手段に与えてキャッシュさせるための選別手段と、プロキシサーバ手段に記録されたファイルオブジェクト名称のリストを放送サーバ装置に送信するためのカテゴリ制御手段とを含む。放送サーバ装置はさらに、リスト送信手段により送信されたファイルオブジェクト名称のリストに基づき、各カテゴリ別の仮想的なキャッシュヒット率を計算してクライアント装置に提示するためのキャッシュヒット率計算手段を含む。カテゴリ制御手段は、キャッシュヒット率計算手段により提示された仮想的なキャッシュヒット率に基づいて、クライアント装置におけるキャッシュヒット率が高くなるように、選別手段により選別されるファイルオブジェクトのカテゴリを設定する。

クライアント装置から送られたファイルオブジェクト名称のリストに基づき、仮想的なキャッシュヒット率が計算され、クライアント装置に提示される。クライアント装置では、この仮想的なキャッシュヒット率に基づいて、選別手段で選別されるカテゴリが、キャッシュヒット率が高くなるように設定される。すなわちユーザのよく利用するカテゴリのファイルオブジェクトがキャッシュされるように、選別手段の選択するカテゴリが設定される。したがって、クライアント装置のプロキシサーバのキャッシュ容量に限度があるときに、キャッシュヒット率を高めることができるように、かつユーザの介入なくキャッシュ内容を選択でき、キャッシュを有効利用できる。またこのカテゴリの設定は、ユーザの利用状況を反映した、プロキシサーバ手段によりキャッシュされたファイルオブジェクト名称のリストの状況に応じて変化するので、ユーザの嗜好が変わっても常にキャッシュヒット率を高く保つことができる。

【0043】請求項2に記載の発明にかかる分散ファイルシステムは、請求項1に記載の分散ファイルシステムであって、放送サーバ装置は、各カテゴリ別のキャッシュサイズをクライアント装置に通知するための手段をさらに含む。カテゴリ制御手段は、放送サーバ装置から通知された各カテゴリ別のキャッシュサイズと、プロキシサーバ装置のキャッシュの割当領域とから、選別手段により選別可能なカテゴリ数を定め、選別手段の選別するカテゴリ数の上限として設定する。

【0044】したがって、請求項1に記載の発明の作用・効果に加えて、クライアント側のキャッシュの容量の上限が予め指定されているときに、このキャッシュを最も有効利用してキャッシュ率を高めることができるように、かつできるだけ幅広いカテゴリの数を定めることができる。

【0045】請求項3に記載の発明にかかる分散ファイルシステムは、請求項1または2に記載の分散ファイルシステムであって、クライアント装置は、特定のカテゴリ番号を記憶する手段をさらに含む。選別手段は、特定のカテゴリ番号のファイルオブジェクトを排除してプロ

キシサーバ手段に与える。

【0046】請求項3に記載の発明によれば、請求項1または2に記載の発明の作用・効果に加えて、特定のカテゴリ番号のファイルオブジェクトが排除されるので、ユーザが望んでいないカテゴリのファイルオブジェクトを受信することが回避され、クライアント装置のキャッシュ用記憶領域を節約できる。

【0047】請求項4に記載の発明にかかる分散ファイルシステムは、請求項1または2に記載の分散ファイルシステムであって、クライアント装置は、特定のカテゴリ番号を記憶する手段をさらに含む。選別手段は、特定のカテゴリ番号のファイルオブジェクトを常に選択してプロキシサーバ手段に与える。

【0048】請求項4に記載の発明によれば、請求項1または2に記載の発明の作用・効果に加えて、特定のカテゴリ番号のファイルオブジェクトが常にキャッシュ対象とされるので、アクセス頻度が低くとも必須のカテゴリのファイルオブジェクトをキャッシュすることができ、クライアント装置のキャッシュの内容をユーザの業務に適したものとすることができる。

【0049】請求項5に記載の発明にかかる分散ファイルシステムは、請求項1に記載の分散ファイルシステムであって、カテゴリ制御手段は、プロキシサーバ手段に記録されたファイルオブジェクト名称のうち、テキストデータに相当するファイルオブジェクト名称のみを選択してファイルオブジェクトのリストとして放送サーバ装置に送信する。

【0050】請求項5に記載の発明によれば、請求項1に記載の発明の効果に加えて、ファイルオブジェクトのリストから、テキストオブジェクト以外のものは排除される。放送サーバ装置に送信されるリストの容量が削減され、放送サーバ装置での処理負荷が軽減される。

【0051】請求項6に記載の発明にかかる分散ファイルシステムは、ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および放送サーバ装置と通信可能なクライアント装置とを含むファイル配送のための分散ファイルシステムである。放送サーバ装置は、サーバ装置から受けたファイルオブジェクトを複数個のカテゴリに分類してキャッシュし、カテゴリ識別情報とともに放送するための放送プロキシサーバ手段を含む。クライアント装置は、キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するとともに、アクセスされたファイルオブジェクト名称をカテゴリ識別情報とともに記録するためのプロキシサーバ手段と、放送サーバ装置により放送されるファイルオブジェクトを受信してプロキシサーバ手段にキャッシュさせるための手段と、プロキシサーバ手段により記録されたファイルオブジェクト名称のリストを放送サーバ装置に送信するための手段とを含む。放送サーバ装置はさらに、送信され

たファイルオブジェクト名称のリストに基づいてサーバ装置からファイルオブジェクトを収集して放送手段に与え、クライアント装置に向けて放送させるための手段を含む。

【0052】この請求項6に記載の発明によれば、放送サーバ装置がサーバ装置から収集するファイルオブジェクトは、各クライアント装置から送信されたファイルオブジェクト名称のリストに基づくものである。また、各リストはカテゴリ識別情報を含み、ファイルオブジェクトはカテゴリ別に蓄積される。そのため、あるカテゴリを選択しているユーザには、別の、同一のカテゴリを選択しているユーザがアクセスしたファイルオブジェクトが放送され、そのクライアント装置にキャッシュされることになり、自己のアクセスのみに限定されず、そのカテゴリの情報をより多くキャッシュすることができる。

【0053】請求項7に記載の分散ファイルシステムは、ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および放送サーバ装置と通信可能で、放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムである。放送サーバ装置は、キャッシュを有し、サーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するための放送用プロキシサーバ手段と、所定の方式にしたがって定められる時刻に、キャッシュされたファイルオブジェクトの、サーバ装置における最終変更時刻情報を放送するための放送手段とを含む。クライアント装置は、キャッシュを有し、サーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理し、アクセスされたファイルオブジェクトの最終変更時刻(HTTPヘッダに含まれるLast-Modified:フィールドの時刻)を記録するためのプロキシサーバ手段と、ユーザによるファイルオブジェクトへのアクセス要求があったときに、プロキシサーバ手段の記録を参照して、当該ファイルオブジェクトのキャッシュファイルの最終ファイル変更時刻と予め定められたキャッシュ有効時間とに基づいて、サーバ装置へのアクセス要求を発行する処理と、キャッシュから当該ファイルオブジェクトを読みだしてユーザに返す処理とのいずれかを選択的に実行するための手段と、放送サーバ装置から受信した最終変更時刻情報に基づき、プロキシサーバ手段により記録された各ファイルオブジェクトの最終変更時刻を更新する処理と、無効と判定されたファイルオブジェクトに対するアクセス要求をプロキシサーバ手段に対して発行する処理とを選択的に実行するための手段とを含む。

【0054】請求項7に記載の発明によれば、各クライアント装置は、実際にファイルオブジェクトにアクセスしなくとも、放送サーバ装置から放送される最終変更時刻情報に基づいて、各キャッシュ内のファイルオブジェクトの有効・無効状況を判定することができる。有効で

あれば最終ファイル変更時刻を受信時刻に更新し、無効であれば改めて当該ファイルオブジェクトをプロキシサーバ手段を通して入手することができ、併せて最終ファイル変更時刻や、キャッシュされたHTTPヘッダのLast-Modified: フィールドの時刻も更新することができる。以後に実際にユーザによるアクセス要求が発生したときに、ファイルオブジェクトが有効と判定される可能性が高くなり、無効と判定されたときのサーバ装置へのアクセスが不要となり、ネットワークのトラフィックの増大を防止でき、またアクセス要求に対するレスポンスを良くすることができる。

【0055】請求項8に記載の発明にかかる分散ファイルシステムは、放送サーバ装置と、放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムである。放送サーバ装置は、ファイルオブジェクトを複数個のカテゴリ別にキャッシュし、カテゴリ識別情報とともに放送するための手段を含む。クライアント装置は、キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するとともに、アクセスされたファイルオブジェクト名称を記録するためのプロキシサーバ手段と、放送サーバ装置によって放送されたファイルオブジェクトを受信し、予め設定されたカテゴリ識別情報を有するファイルオブジェクトのみを選別してプロキシサーバ手段に与えカテゴリ識別情報とともにキャッシュさせるための選別手段と、プロキシサーバ手段によって処理された最新のカテゴリ識別情報を記憶するための手段と、プロキシサーバ手段に記録されたファイルオブジェクト名称のカテゴリ別リストを、放送サーバ装置に送信するためのカテゴリ別リスト送信手段と、プロキシサーバ手段によるサーバ装置へのアクセスがあったときに、アクセスされたファイルオブジェクトのカテゴリ識別情報として記憶された最新のカテゴリ識別情報を設定するための手段とを含む。放送サーバ装置はさらに、カテゴリ別リスト送信手段により送信されたファイルオブジェクト名称のカテゴリ別リストに基づいてサーバ装置からファイルオブジェクトを収集して放送手段に与え、クライアント装置に向けて放送させてクライアント装置のキャッシュにカテゴリ別に蓄積させるための手段を含む。

【0056】請求項8に記載の発明によれば、クライアントがキャッシュされていないファイルオブジェクトをサーバ装置にアクセスして入手したとき、プロキシサーバ手段による記録の際のカテゴリ識別情報として、その直前にアクセスされた最新のファイルオブジェクトのカテゴリ識別情報に設定する。一般に、ユーザは同一のカテゴリに属するファイルオブジェクトをアクセスする傾向があるので、このようにすることで新規にキャッシュされたファイルオブジェクトのカテゴリを適切に設定できる。そうして設定されたカテゴリ識別情報が放送サー

バ装置に送信され、ファイルオブジェクトのカテゴリ別の蓄積および放送に使用されるので、各クライアント装置において、対応のカテゴリのファイルオブジェクトのキャッシュヒット率を高くすることができる。

【0057】請求項9に記載の発明にかかる放送サーバ装置は、ファイルオブジェクトを複数個のカテゴリに分類してキャッシュし、カテゴリ識別情報とともに放送するための放送用プロキシサーバ手段と、クライアント装置から送信されてくる、クライアント装置によってアクセスされたファイルオブジェクト名称のリストに基づき、クライアント装置における各カテゴリ別の仮想的なキャッシュヒット率を計算してクライアント装置に提示するためのキャッシュヒット率計算手段とを含む。

【0058】請求項9に記載の発明によれば、こうして計算された仮想的なキャッシュヒット率を用いて、クライアント装置側で適切なカテゴリを選択することができる。したがって、クライアント装置におけるキャッシュヒット率を高めることが可能になる。

【0059】請求項10に記載の発明にかかる放送サーバ装置は、請求項9に記載の放送サーバ装置であって、各カテゴリ別のキャッシュサイズをクライアント装置に通知するための手段をさらに含む。

【0060】請求項10に記載の発明によれば、クライアント装置側では、放送サーバ装置側から通知されが各カテゴリ別のキャッシュサイズに基づき、キャッシュ対象として選別するカテゴリの数を、自己のキャッシュ割当領域に合わせて設定できる。そのため、各クライアント装置は、キャッシュヒット率を高めながら、できるだけ広い範囲のカテゴリのファイルオブジェクトをキャッシュすることができる。

【0061】請求項11に記載の発明にかかる放送サーバ装置は、サーバ装置から受けたファイルオブジェクトを複数個のカテゴリに分類してキャッシュし、カテゴリ識別情報とともに放送するための放送プロキシサーバ手段と、クライアント装置から送信された、クライアント装置がアクセスしたファイルオブジェクト名称のリストに基づいてサーバ装置からファイルオブジェクトを収集して放送手段に与え、クライアント装置に向けて放送させるための手段とを含む。

【0062】請求項11に記載の発明によれば、放送サーバ装置は、クライアント装置のアクセスした記録を反映したリストに基づいてファイルオブジェクトを収集し放送する。実際の各クライアント装置によるファイルオブジェクトのアクセスを反映した形でファイルオブジェクトが各クライアント装置にキャッシュされるので、各クライアント装置におけるキャッシュヒット率を高くすることができる。

【0063】請求項12に記載の発明にかかる放送サーバ装置は、キャッシュを有し、サーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理し

て処理するための放送用プロキシサーバ手段と、所定の方式にしたがって定められる時刻に、キャッシュされたファイルオブジェクトの、サーバ装置における最終変更時刻情報を放送するための放送手段とを含む。

【0064】請求項12に記載の発明によれば、放送用プロキシサーバ手段がアクセスしキャッシュしたファイルオブジェクトの最終変更時刻が、各クライアント装置に対して放送される。各クライアント装置では、自己のキャッシュしているファイルオブジェクトに含まれるLast-Modified:フィールドの最終変更時刻と、放送された最終変更時刻とを照合し、一致するならばその受信された時刻を以て最終変更時刻に更新することができる。すなわち最終変更時刻の確認を、受信時にすることで、放送プロキシサーバにおいて最新性が確認された時刻からわずかな遅延を以て一致させることができる。正確には放送プロキシサーバにおいて最新性が確認された時刻と受信時刻との間にはずれが生ずるが、その場合は放送時に最新性確認時刻を含めて放送し、受信時にその時刻を最終ファイル変更時刻とすれば正確である。一致しないときには、すなわちサーバ装置のファイルオブジェクトの最終変更時刻の方が新しければ、改めてプロキシサーバ手段を介してそのファイルオブジェクトを入手でき、併せてその最終変更時刻を最新のものに更新できる。そのクライアント装置において以後にそのファイルオブジェクトファイルへのアクセス要求が発生したときに、そのファイルオブジェクトが有効と判定される可能性が高くなる。無効と判定されたときに、当該ファイルオブジェクトを入手するためにサーバ装置に対してアクセスを行なう必要がなくなる。その結果、ネットワークのトラフィックを軽減できるとともに、アクセス要求に対するレスポンスを速くすることができる。

【0065】請求項13に記載の発明にかかる放送サーバ装置は、ファイルオブジェクトを複数個のカテゴリ別にキャッシュし、カテゴリ識別情報とともに放送するための手段と、クライアント装置から送信された、クライアント装置によってアクセスされたファイルオブジェクト名称のカテゴリ別リストに基づいてサーバ装置からファイルオブジェクトを収集して放送手段に与え、クライアント装置に向けて放送させてクライアント装置のキャッシュにカテゴリ別に蓄積させるための手段とを含む。

【0066】請求項13に記載の発明によれば、クライアント装置によるアクセス結果を反映したファイルオブジェクトが収集され、放送サーバ装置によって放送される。各クライアントでは、自己のアクセス結果を反映したファイルオブジェクトをさらにキャッシュすることができ、それぞれのキャッシュのキャッシュヒット率を高く維持することができる。また、複数のクライアント装置からのカテゴリ別リストに基づいてファイルオブジェクトが収集されるので、あるカテゴリを選択したユーザのクライアント装置のキャッシュには、そのカテゴリを

選択した他のユーザのアクセス結果に基づいたファイルオブジェクトもキャッシュされることになり、キャッシュのヒット率を高めることができる。

【0067】請求項14に記載の発明にかかるクライアント装置は、キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するためのプロキシサーバ手段と、放送サーバ装置によって放送されたファイルオブジェクトを受信し、予め設定されたカテゴリのファイルオブジェクトのみを選別してプロキシサーバ手段に与えてキャッシュさせるための選別手段と、プロキシサーバ手段に記録されたファイルオブジェクト名称のリストを放送サーバ装置に送信するためのカテゴリ制御手段とを含む。放送サーバ装置は、カテゴリ制御手段により送信されたファイルオブジェクト名称のリストに基づき、各カテゴリ別の仮想的なキャッシュヒット率を計算してクライアント装置に提示するためのキャッシュヒット率計算手段を含む。カテゴリ制御手段は、放送サーバ装置のキャッシュヒット率計算手段により提示された仮想的なキャッシュヒット率に基づいて、クライアント装置におけるキャッシュヒット率が高くなるように、選別手段により選別されるファイルオブジェクトのカテゴリを設定する。

【0068】請求項14に記載の発明によれば、放送サーバ装置で計算された仮想的なキャッシュヒット率に基づいて、クライアント装置で選別されるファイルオブジェクトのカテゴリを、キャッシュヒット率が高くなるように設定することができる。

【0069】請求項15に記載の発明にかかるクライアント装置は、請求項14に記載のクライアント装置であって、放送サーバ装置は、各カテゴリ別のキャッシュサイズをクライアント装置に通知するための手段をさらに含む。カテゴリ制御手段は、放送サーバ装置から通知された各カテゴリ別のキャッシュサイズと、プロキシサーバ装置のキャッシュの割当領域とから、選別手段により選別可能なカテゴリ数を定め、選別手段の選別するカテゴリ数の上限として設定する。

【0070】請求項15に記載の発明によれば、請求項14に記載の発明の作用・効果に加え、クライアント装置は、プロキシサーバのカテゴリ別のキャッシュ容量に基づいて、選別手段のカテゴリ数の上限を設定することができる。そのため、限定されたキャッシュ容量の制限のもとで、キャッシュヒット率を高めながら、できるかぎり幅広いカテゴリのファイルオブジェクトをキャッシュすることが可能になる。

【0071】請求項16に記載の発明にかかるクライアント装置は、請求項14または15に記載のクライアント装置であって、特定のカテゴリ番号を記憶する手段をさらに含み、選別手段は、特定のカテゴリ番号のファイルオブジェクトを排除してプロキシサーバ手段に与え

る。

【0072】請求項16に記載の発明によれば、請求項14または15に記載の発明の作用・効果に加え、ユーザが望まないカテゴリファイルオブジェクトをキャッシュすることを回避でき、キャッシュ領域を有効利用できる。

【0073】請求項17に記載の発明にかかるクライアント装置は、請求項14または15に記載のクライアント装置であって、クライアント装置は、特定のカテゴリ番号を記憶する手段をさらに含む。選択手段は、特定のカテゴリ番号のファイルオブジェクトを常に選択してプロキシサーバ手段に与える。

【0074】請求項17に記載の発明によれば、請求項14または15に記載の発明の作用・効果に加え、アクセス頻度が低くとも必須のカテゴリのファイルオブジェクトをキャッシュしておくことができ、キャッシュヒット率を高めながら、ユーザのクライアント装置のキャッシュファイルの内容をそのユーザの業務に適したものに維持することができる。

【0075】請求項18に記載の発明にかかるクライアント装置は、請求項14に記載のクライアント装置であって、カテゴリ制御手段は、プロキシサーバ手段に記録されたファイルオブジェクト名称のうち、テキストデータに相当するファイルオブジェクト名称のみを選択してファイルオブジェクト名称のリストとして送信するための手段を含む。

【0076】請求項18に記載の発明によれば、請求項14に記載の発明の作用・効果に加え、テキストデータに相当するファイルオブジェクト名称のみが選択されて放送サーバ装置に送信され、他の種類のファイルオブジェクト名称は送信の対象から除外される。放送サーバ装置ではこの送信されたファイルオブジェクト名称に基づいて処理を行なうため、テキストデータ以外のものも含めて送信する場合と比較して放送サーバ装置における処理負荷が軽減される。

【0077】請求項19に記載の発明にかかるクライアント装置は、ネットワーク上のサーバ装置と通信可能な放送サーバ装置と、サーバ装置および放送サーバ装置と通信可能なクライアント装置とを含むファイル配送のための分散ファイルシステムにおいて用いられるクライアント装置である。放送サーバ装置は、サーバ装置から受けたファイルオブジェクトを複数個のカテゴリに分類してキャッシュし、カテゴリ識別情報とともに放送するための放送プロキシサーバ手段を含む。クライアント装置は、キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するとともに、アクセスされたファイルオブジェクト名称を記録するためのプロキシサーバ手段と、放送サーバ装置により放送されるファイルオブジェクトを受信してプロキシサーバ手段にキャッシュさせる

ための手段と、プロキシサーバ手段により記録されたファイルオブジェクト名称のリストを放送サーバ装置に送信するための手段とを含む。

【0078】請求項19に記載の発明によれば、クライアント装置のプロキシサーバ手段により記録されたファイルオブジェクト名称が放送サーバ装置に送信される。放送サーバ装置はこの送信されたファイルオブジェクト名称に基づいてサーバ装置からファイルオブジェクトを収集してクライアント装置に対して放送する処理を行なうことができる。そのため、各クライアントのキャッシュには、各クライアントにおけるアクセス結果を反映したファイルオブジェクトが蓄積されることになり、キャッシュヒット率を高めることができる。

【0079】請求項20に記載の発明にかかるクライアント装置は、キャッシュを有し、サーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するための放送用プロキシサーバ手段と、所定の方式にしたがって定められる時刻に、キャッシュされたファイルオブジェクトの、サーバ装置における最終変更時刻情報を放送するための放送手段とを含む放送サーバ装置と通信可能である。クライアント装置は、キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するとともに、アクセスされたファイルオブジェクト名称の最終変更時刻を記録するためのプロキシサーバ手段と、ユーザによるファイルオブジェクトへのアクセス要求があったときに、プロキシサーバ手段の記録を参照して、当該ファイルオブジェクトの最終ファイル変更時刻と予め定められたキャッシュ有効時間とに基づいて、サーバ装置へのアクセス要求を放送サーバ装置に発行する処理と、キャッシュから当該ファイルオブジェクトを読みだしてユーザに返す処理とのいずれかを選択するための手段と、放送サーバ装置から受信した最終変更時刻情報に基づき、プロキシサーバ手段により記録された各ファイルオブジェクトの最終ファイル変更時刻を更新する処理と、無効と判定されたファイルオブジェクトに対するアクセス要求をプロキシサーバ手段に対して発行する処理とを選択的に実行するための手段とを含む。

【0080】請求項20に記載の発明によると、放送サーバ装置から送信された各ファイルオブジェクトの最終変更時刻に基づいて、各クライアントにおいてキャッシュされているファイルオブジェクトの最終ファイル変更時刻が更新される。以後にこのファイルオブジェクトに対するアクセス要求が発生したときに、最終ファイル変更時刻と予め定められたキャッシュ有効時間とに基づいて、当該ファイルオブジェクトが有効かどうか判定される。最終変更時刻が、放送サーバ装置から最終変更時刻が放送されたときに更新されているので、実際のアクセス要求の際にキャッシュされているファイルオブジェクトが有効とされる確率が高くなり、その結果あらため

てファイルオブジェクトを入手するためにサーバ装置に対してアクセスする必要が生ずる頻度が低くなる。その結果、ネットワークのトラフィックの増大を防止でき、またアクセス要求に対するレスポンスを速くすることができる。

【0081】請求項21に記載の発明にかかるクライアント装置は、放送サーバ装置と、放送サーバ装置により放送されるファイルオブジェクトを受けるクライアント装置とを含むファイル配送のための分散ファイルシステムにおいて用いられるクライアント装置である。放送サーバ装置は、ファイルオブジェクトを複数のカテゴリ別にキャッシュし、カテゴリ識別情報とともに放送するための手段を含む。このクライアント装置は、キャッシュを有し、ユーザによるサーバ装置のファイルオブジェクトに対するアクセス要求をサーバ装置を代理して処理するとともに、アクセスされたファイルオブジェクト名称を記録するためのプロキシサーバ手段と、放送サーバ装置によって放送されたファイルオブジェクトを受信し、予め設定されたカテゴリ識別情報を有するファイルオブジェクトのみを選択してプロキシサーバ手段にカテゴリ識別情報とともに与えてキャッシュさせるための選別手段と、プロキシサーバ手段により処理された最新のカテゴリ識別情報を記憶するための手段と、プロキシサーバ手段に記録されたファイルオブジェクト名称のカテゴリ別リストを、放送サーバ装置に送信するための手段と、プロキシサーバ手段によるサーバ装置へのアクセスがあったときに、アクセスされたファイルオブジェクト

http://www.xxx.co.jp/test/index.html … (3)
 {<1>}<—2—>}<—3—>}

このURLのうち、「1」で示される「http」の部分は使用するプロトコルを特定する。「2」で示される「www.xxx.co.jp」はネットワーク上のHTTPサーバ計算機のアドレスを示すものであり、ネットワーク上でユニークに選ばれている。「3」で示される「/test/index.html」はサーバ計算機のファイル名称を示す。

【0085】proxyプロセスはファイルオブジェクトの中継記録としてコモンログフォーマットと呼ばれる形式

```
クライアント計算機名称 -- [時刻] "GET http://naragw.sharp.co.jp/oldindex.html HTTP/1.0" 200 1717
クライアント計算機名称 -- [時刻] "GET http://naragw.sharp.co.jp/sharpcolor.gif HTTP/1.0" 200 1370
クライアント計算機名称 -- [時刻] "GET http://naragw.sharp.co.jp/hsg.mono.gif HTTP/1.0" 200 3331
クライアント計算機名称 -- [時刻] "GET http://naragw.sharp.co.jp/Shoin.gif HTTP/1.0" 200 17936
クライアント計算機名称 -- [時刻] "GET http://naragw.sharp.co.jp/Zaurus.gif HTTP/1.0" 200 16859
クライアント計算機名称 -- [時刻] "GET http://naragw.sharp.co.jp/Prostation.gif HTTP/1.0" 200 16345
クライアント計算機名称 -- [時刻] "GET http://naragw.sharp.co.jp/hsg.gif HTTP/1.0" 200 20782
クライアント計算機名称 -- [時刻] "GET http://naragw.sharp.co.jp/S2.gif HTTP/1.0" 200 11635
```

【0087】サーバ計算機の状態コードとは、サーバ計算機がクライアント計算機からの要求に対して発する三桁の数字で表される応答である。例えば、200の場合は、要求が正常に受け付けられた旨を示す。状態コードの最初の数字はカテゴリを示しており、1、2、3、4、

のカテゴリ識別情報として記憶された最新のカテゴリ識別情報を設定するための手段とを含む。

【0082】請求項21に記載の発明によれば、クライアントがキャッシュされていないファイルオブジェクトをサーバ装置にアクセスして入手したとき、プロキシサーバ手段による記録の際のカテゴリ識別情報として、その直前にアクセスされた最新のカテゴリ識別情報が設定される。一般に、ユーザは同一のカテゴリに属するファイルオブジェクトをアクセスする傾向があるので、このようにすることで新規にキャッシュされたファイルオブジェクトのカテゴリを適切に設定できる。そうして設定されたカテゴリ識別情報が放送サーバ装置に送信され、ファイルオブジェクトのカテゴリ別の蓄積および放送に使用されるので、各クライアント装置において、対応のカテゴリのファイルオブジェクトのキャッシュヒット率を高くすることができる。

【0083】

【発明の実施の形態】

【実施の形態1】実施の形態1の詳細を述べるに先立ち、まず用語を定義する。この発明の主な適用分野として既に述べたようにWWWシステムがある。WWWシステムではネットワーク上に分散したファイルオブジェクト名称はUniform Resource Locator(URL)とよばれる形式で表現され、特定される。URLの例を以下に示す。

【0084】

【数3】

に従ったアクセスログ(ファイルオブジェクト転送記録)を生成する。アクセスログは、内部ネットワークのクライアント計算機名称、時刻、“HTTPメソッド(ファイル取得要求)”、サーバ計算機の状態コード、ファイルオブジェクトのデータサイズ等を含む。標準的なアクセスログ形式を以下の表1に示す。

【0086】

【表1】

5 が使用される。200番台の状態コードはクライアント計算機の要求を問題なく理解したことを、300番台の状態コードはサーバ計算機からファイルオブジェクトの再転送の必要があることを、400番台の状態コードは存在しないURLをクライアント計算機がアクセスした場合の

エラーを、500番台の状態コードはサーバ計算機のエラーを、それぞれ表わしている。100番台の状態コードは将来のために予約されており、現在は使用されていない。

【0088】ファイルオブジェクトのデータサイズは、ファイルオブジェクトのバイト数を示す。

【0089】図1を参照して、本実施の形態にかかる分散ファイルシステム20は、放送サーバ22と、複数のクライアント計算機26、28等と、ネットワーク上に接続された多数のWebサーバ24とを含む。

【0090】放送サーバ22は、ファイルオブジェクトを後述するカテゴリ別にキャッシュするための複数のキャッシュファイルシステム62と、放送Proxyサーバ部64と、キャッシュ送出制御部70と、クライアント計算機26、28等からアクセスログを受信し、各クライアントに対する仮想キャッシュヒット率を計算する仮想キャッシュヒット率算出部80を持つアクセスログ受信用WWWサーバ60と、キャッシュ送出制御部70が送出するファイルオブジェクトのリストである放送リスト68と、キャッシュ送出制御部70から与えられるファイルオブジェクトを各クライアントに向けて放送するための放送ハードウェア72とを含む。

【0091】クライアント計算機26、28等は基本的に互いに同じ構成を有する。たとえばクライアント計算機26は、カテゴリ制御部50と、キャッシュ受信部48と、クライアントProxyサーバ部46と、そのキャッシュファイル42と、アクセスログ44と、カテゴリ制御部50が使用するワークファイル56と、デジタル衛星放送の受信ボードまたはネットワークカード等の放送受信ハードウェア部52と、ブラウザ40とを含む。ブラウザ40は、クライアント計算機26で動作していてもよいし、ローカルネットワークで結ばれた他のクライアント計算機において動作してもよい。ブラウザ40はクライアントProxy 46を経由してサーバ計算機24にアクセスする。

【0092】クライアント計算機としては、具体的にはパーソナルコンピュータで、1GB程度のディスク装置とモデム等の比較的低速なネットワーク手段とを持っているものが想定される。現在パーソナルコンピュータのディスク装置は1GB以上が普通になってきており今後もディスク装置の容量は増加するものと思われる。また工場出荷時にあらかじめOS（オペレーティング・システム）やアプリケーション・ソフトウェア等がディスクに格納されるようになっているが、それでもディスク装置の未使用領域は500MB以上存在する。したがって前述のようにクライアント計算機26のキャッシュファイル42

として300MB程度の容量を確保することは困難ではない。

【0093】クライアントProxy 46においては、ファイルオブジェクトはクライアント計算機26のファイルシステムに適合した形で格納される必要がある。たとえばhttp://www.cnn.com/index.htmlというファイルオブジェクトは、クライアント計算機26のディスク装置（仮にCドライブとする。）のC:\%cache%\http\www.cnn.com\index.htmlというファイルに変換してキャッシュされるものとする。このようにクライアントProxy 46によってキャッシュされたファイルオブジェクト全体を以下「クライアントキャッシュ」と呼ぶ。また、クライアントキャッシュのうちの個々のファイルオブジェクトも同じ名称で呼ぶことがある。

【0094】本実施の形態では、クライアントキャッシュのファイルオブジェクトの記録としてのアクセスログ44がクライアント計算機26、28等から放送サーバ計算機22に送信され、放送サーバ計算機22がこれらアクセスログに基づいてWebサーバ24からファイルオブジェクトを取得してキャッシュファイルシステム62にキャッシュし、これらファイルオブジェクトを複数のクライアント計算機に対して放送する。各クライアント計算機26、28等は、放送されたファイルオブジェクトをキャッシュファイル42に格納することでクライアントキャッシュを増大させてヒット率を向上させ、Webサーバ計算機24へのアクセスを見かけ上高速化する。

【0095】放送サーバ22における放送用Proxyサーバプロセス64は、ファイルオブジェクトをカテゴリごとにキャッシュファイルシステム62に分類し保持する。カテゴリは、各カテゴリごとのキャッシュの総量が100MB程度になるように分類されている。既に述べた、現在利用可能なProxyソフトウェアでは、キャッシュサイズの上限を指定することができる。したがってカテゴリのこのような細分化は既存の技術を適切に利用することで実現できる。本実施の形態および以下に述べる他の実施の形態においては、放送サーバ計算機22はUNIXをOSとする計算機であるものとする。もちろん、放送サーバ計算機がUNIXをOSとする計算機に限定されるわけではなく、適切なソフトウェアが動作する、ネットワークインタフェースを有するものであればどのようなものでもよい。

【0096】たとえば放送サーバ計算機22におけるカテゴリとディレクトリとは以下の表2に示すように細分化される。

【0097】

【表2】

ID	カテゴリ	キャッシュ directory
ID=0	ローカルニュース	/cache/0/
ID=1	国内ニュース	/cache/1/
ID=2	海外ニュース	/cache/2/
ID=3	コンピュータ業界ニュース	/cache/3/
ID=4	ビジネス情報	/cache/4/
ID=5	ホテル情報	/cache/5/
ID=6	旅行情報	/cache/6/
ID=7	ローカル天気情報	/cache/7/
ID=8	世界の天気情報	/cache/8/
ID=9	日本の天気情報	/cache/9/

【0098】なお、カテゴリごとにWebサーバ計算機24から情報を収集する方法としては、たとえば、オフラインリーダソフトウェアと呼ばれる市販のソフトウェアを使って特定URL以下の階層を定期的に放送用Proxyサーバ64経由でアクセスする等の方法がある。オフラインリーダソフトウェアとしてはForeFront社の商品名WebWhackerやXaxon社の商品名NetRecorder等がある。

【0099】キャッシュファイルシステム62の各カテゴリのカテゴリサイズは、放送サーバ計算機22で運用するWWWサーバ60等を通じてクライアント計算機のメモリに通知することができる。なお、この通知をするサーバプロセスはWWWサーバ60とは独立したものであってもよい。したがって、クライアント計算機26、28等のキャッシュ用割り当て領域がユーザにより指定されていれば、その領域の大きさと、通知された各カテゴリのキャッシュサイズとに基づいて、何個のカテゴリを受信するかをクライアント計算機側で決定できる。

【0100】放送サーバ側でのキャッシュ形式は以下の通りである。なお以下の説明では、ファイルオブジェクトがキャッシュディレクトリ名を先頭として「キャッシュ

directory 名/ カテゴリ番号/ プロトコル/ サーバ計算機名称/URLファイル部名称」なるファイルパスおよびファイル名でキャッシュされるものとする。例えば「http://www.sharp.co.jp/news/index.html」というファイルオブジェクトは「/cache/3/http/www.sharp.co.jp/news/index.html」等としてキャッシュされる。ここで、「/cache/3/」の「3」はカテゴリIDである。

【0101】たとえばhttp://www.asahi.com/等の新聞社のサーバ計算機のオブジェクトは国内ニュース、海外ニュース等に分類されてキャッシュされる。これらの分類は人為的に行う方法と自動化する方法とがある。自動化する方法については実施の形態2で説明する。

【0102】キャッシュされたファイルオブジェクトの実体は、サーバ計算機からのHTTP応答ヘッダとボディ部とを一体にしてファイルにしたものである。例えば、「/cache/3/http/www.sharp.co.jp/news/index.html」ではファイルオブジェクトは以下の形式である。

【0103】

【表3】

```

HTTP/1.0 200 OK¥r¥n
Server: Netscape-Commerce/1.12¥r¥n
Date: Friday, 30-May-97 07:00:09 GMT¥r¥n
Last-Modified: Friday, 30-May-97 01:03:52 GMT¥r¥n
Content-Length: 10673¥r¥n
Content-type: text/html¥r¥n
¥r¥n
<html>¥r¥n
<head><title>News Page</title></head>¥r¥n
<body>¥r¥n
<h1>Whats New Tday</h1>¥r¥n

```

【0104】ここで「¥r¥n」は改行を意味する。HTTP応答ヘッダは1行目から6行目までであり、その後は¥r¥nの空行で分離されたボディ部が続く。ここではボディ部はHTML (HyperText Markup Language)で記述されたテキストとなっている。

【0105】放送サーバ計算機22のキャッシュ送出制御部70は以下のように動作する。キャッシュ送出制御部70は、一定の時間間隔ごとに、またはスケジュールされた時刻になる等一定の方式にしたがって定められたタイミングになると、以下の処理手順に従ってキャッシュ

ュファイル62内のファイルオブジェクトを放送する。なお各カテゴリごとにキャッシュ送出のための処理プロセスが存在し、それぞれのカテゴリごとの処理を並行して行うものとする。ここではそのうちのひとつのプロセスの処理を示す。

【0106】一回の放送処理に対しては、予めその終了時刻と、最大放送バイト量MAX[id]とを指定しておく。ここで「id」とあるのは、カテゴリの識別番号である。すなわち、各カテゴリ別に最大放送バイト量が定められる。

```
cd /cache/1/http/; /bin/find . -print > list1.txt ... (4)
```

これにより放送リスト68（ファイル名を「list1.txt」とする。）には次のようなファイル名称がリストされる。

```
www.sharp.co.jp/news/index.html
www.sharp.co.jp/news/image/1.gif
www.sharp.co.jp/news/image/2.gif
```

【0110】次にファイルリストをもとに放送を行う前準備としてヘッダ付けを行う（92）。

【0111】放送に際しては放送リスト68を読み出し、読み出し位置の情報に対応するファイルをオープンし、カテゴリIDとURL 名ヘッダとをHTTP応答ヘッダ部に付加する。カテゴリIDはX-CategoryID:、URL ヘッダはX-URL: というヘッダを識別子として使用する。URL ヘッダは、キャッシュ放送されるファイルオブジェクトがどの

【0107】図2を参照して、キャッシュ送出制御部70は、カテゴリの放送リスト68をまず作成する（ステップ90。以下「ステップ」の記載は省略する。）。例えば放送サーバ計算機22がUNIX OS採用の計算機であれば第1のキャッシュカテゴリの全てのファイルオブジェクトの名称をリストにするには、ファイルリストを再帰的に出力するfindコマンドを使って次のようなコマンドをOSに対して発行する。

【0108】

【数4】

【0109】

【表4】

URL オブジェクトかを識別するために使用するもので、キャッシュファイル名からURL を復元する。

【0112】すなわち「/cache/3/http/www.sharp.co.jp/news/index.html」というファイルに関しては放送用データの構造は以下のようになっている。

【0113】

【表5】

```
HTTP/1.0 200 OK¥r¥n
X-CategoryID: 3¥r¥n
X-URL: http://www.sharp.co.jp/news/index.html¥r¥n
Server: Netscape-Commerce/1.12¥r¥n
Date: Friday, 30-May-97 07:00:09 GMT¥r¥n
Last-Modified: Friday, 30-May-97 01:03:52 GMT¥r¥n
Content-Length: 10673¥r¥n
Content-type: text/html¥r¥n
¥r¥n
<html>¥r¥n
<head><title>News Page</title></head>¥r¥n
<body>¥r¥n
<h1>Whats New Tbdays</h1>¥r¥n
```

【0114】放送ハードウェア72に送出されるのはこのデータである。続いて、送出バイト数の上限のチェックを行なうためのメモリ変数に以下に示されるように累

```
Total[CategoryID] += このファイルオブジェクトのバイト数 ... (5)
```

放送ヘッダを付加したファイルオブジェクトを放送ハードウェア部72に送り出す（94）。放送ハードウェア部72は、指定ファイルオブジェクトをクライアント計算機26、28等までの経路に応じて送出する。放送ハ

積放送バイト数を加算する。

【0115】

【数5】

ードウェア部72は、この実施の形態ではデジタル衛星放送を利用したり有線LAN のMulticastIPを利用したりして放送する。

【0116】クライアント計算機26、28等では、受

信ハードウェア部52、54等により放送サーバ22により送出されたファイルオブジェクトを受信できる。

【0117】ふたたび図2を参照して、指定時刻が来るか、または、累積放送バイト数が指定バイト数を超えるかした場合は終了する（ステップ96において「yes」と判定された場合）。

【0118】以上の送出フローによりキャッシュカテゴリIDとURL 名称とが付加されたファイルオブジェクトが放送された。なお、既に述べたように、カテゴリ毎に上記放送プロセスが並行動作する。

【0119】クライアント計算機は以下の様に動作する。例としてクライアント計算機26について説明する。クライアント計算機26では放送キャッシュ受信部48と、キャッシュ付きクライアントProxy サーバ46とが常時動いている。ユーザはこのクライアントProxy サーバ46経由でWeb サーバ計算機24をアクセスするようにブラウザ40を設定して利用する。Proxy サーバ46は、ユーザによるアクセス要求がブラウザ40から与えられると、キャッシュ42に当該ファイルオブジェクトがあるかどうか調べ、なければネットワークを経由してファイルオブジェクトをWeb サーバ計算機24から取得し、キャッシュファイル42にキャッシュするとともにブラウザ40にファイルオブジェクトを返す。このとき、アクセスログ44にアクセスログが記録される。

【0120】一方、放送サーバ22が放送したファイルオブジェクトは、クライアント計算機26の受信ハードウェア部52により受信されキャッシュ受信部48に与えられる。キャッシュ受信部48は、受信したファイルオブジェクトのうち、予め指定されたカテゴリ（カテゴリ制御部50が保持するカテゴリリストにより決定される。）のもののみをクライアントProxy サーバプロセス46のキャッシュ42に引き渡す。

【0121】カテゴリ制御部50は、受信カテゴリを決定するカテゴリリストCategoryID[] 配列をメモリに保持する。またカテゴリ制御部50は、後述する仮想ヒット率を保持するテーブルHit[]、MisHit[] 配列を各CategoryIDごとに持つ。

【0122】図3を参照して、キャッシュ受信部50は以下のように動作する。まず、キャッシュ受信部48の初期化処理として、CategoryIDを初期化する（100）。これはユーザにGraphics User Interface を使っ

た入力画面を使って決めさせてもよい。また、ランダムに複数選んでもよい。

【0123】仮に初期化されたCategoryID[] = {0,5,8} とする。カテゴリの数は有限としここでは3個とする。前述したとおり、この配列内に保持されたカテゴリのファイルオブジェクトしかクライアント計算機26のキャッシュ42にキャッシュされない。これによりクライアント計算機26のキャッシュファイル42の容量として必要なのは、カテゴリごとのキャッシュサイズの合計となる。したがって、カテゴリ分類をしないでキャッシュする場合と比較して少ない容量のキャッシュでよい。例えば各カテゴリ100MBであればクライアント計算機26のキャッシュ容量の合計は300MBとなる。

【0124】次に、キャッシュ受信部48は受信ハードウェア部52を監視するループ処理に入る。受信ハードウェア部52を監視してデータ受信があれば、キャッシュ受信部48はまず、受信されたファイルオブジェクトのヘッダを読み、そのX-CategoryID: フィールドの数字、すなわちカテゴリ識別番号とカテゴリ制御部50の保持するCategoryID[] 配列とを比較する。両者が一致した場合は、受信したファイルオブジェクトをクライアントProxy プロセス46の管理するキャッシュとしてキャッシュファイル42に蓄積する（102）。そうでない場合は捨てる。

【0125】このとき、受信したファイルオブジェクトの識別にはファイルオブジェクトのHTTPヘッダに埋め込まれたX-URL:ヘッダを利用する。たとえば、受信したファイルオブジェクトのX-URL ヘッダが「X-URL: http://www.sharp.co.jp/news/index.html」であればこのファイルオブジェクトを「C:\\$cache¥http¥www.sharp.co.jp¥news¥index.html」にコピーすることによってクライアントProxy 46のキャッシュとすることができる。

【0126】以後、このループ処理を繰返す。なお、クライアントProxy サーバプロセス46のアクセスログは、前述した表1の標準形式にカテゴリIDを追加した形式であるものとする。アクセスログ44にはブラウザ40経由でアクセスされたものだけ記録する。クライアントProxy のアクセスログ形式を以下に示す。

【0127】

【表6】

```

クライアント計算機名称 -- [時刻] "GET http://naragw.sharp.co.jp/oldindex.html HTTP/1.0" 200 1717 0
クライアント計算機名称 -- [時刻] "GET http://naragw.sharp.co.jp/sharpcolor.gif HTTP/1.0" 200 1370 2
クライアント計算機名称 -- [時刻] "GET http://naragw.sharp.co.jp/isg.mono.gif HTTP/1.0" 200 3331 5

```

【0128】カテゴリIDは受信キャッシュオブジェクトのX-CategoryID: から取り出した情報である。なおユーザがアクセスしたURL がキャッシュにない場合、そのファイルオブジェクトはサーバ計算機24のうちの該当するものから取得することになる。こうして取得されたフ

ァイルオブジェクトはX-CategoryID: フィールドを持たない。そのため、そのままではこのファイルオブジェクトをカテゴリに分類することができない。このときには、その直前のURL のカテゴリIDを使用する。すなわち、直前にアクセスしたファイルオブジェクトのカテゴリ

リIDを変数Categorynow に記憶し、新たに取得されたファイルオブジェクトのカテゴリとする。

【0129】このようにしても、新たに取得されたファイルオブジェクトが正しいカテゴリに分類される可能性は非常に高い。これはブラウザを使ってユーザのアクセスしているURL のカテゴリは徐々にしか変化しない、という現象に由来する。例えばニュース系のURL をアクセスし始めればハイパーリンクをたどりながら新しいURL をブラウジングしてゆくが、これらはいずれもニュースカテゴリに属することが多いであろう。したがってあるファイルオブジェクトの属するカテゴリは、その直前にアクセスされたファイルオブジェクトの属するカテゴリと高い確率で等しくなることが期待できる。これは、ニュース系に限らず他のカテゴリの場合についても同様である。

【0130】ここで、もし直前のURL のカテゴリIDが不定であれば、止むを得ないのでCategoryID[]配列の最初の要素をそのカテゴリとする。すなわち次の処理をする。

【0131】

【数6】Categorynow = CategoryID[0]; ... (6)
表6に掲げた例では、各ファイルオブジェクトの属するカテゴリのカテゴリIDはそれぞれ0、2、5である。

【0132】こうして受信されたファイルオブジェクトを、各クライアント計算機のクライアントProxy 46のキャッシュとして使用できると、各クライアント計算機におけるそれらURL へのアクセスを高速化できる。

【0133】カテゴリ制御部50は以下のように動作する。カテゴリ制御部50はアクセスログ44を監視し、一定以上蓄積されればそれを放送サーバ計算機22に通知する。これを受信した放送サーバ計算機22は、後述するようにそのクライアント計算機に対するカテゴリごとの仮想キャッシュヒット率を計算し当該クライアントに送信してくる。カテゴリ制御部50は、こうして放送サーバ計算機22から送信されてくるカテゴリ毎の仮想キャッシュヒット率を受信し、そのクライアント計算機

においてヒット率が最も高くなるように、カテゴリID配列に記憶されるカテゴリIDを変化させる。

【0134】図4を参照して、カテゴリ制御部50で行なわれる処理手順は以下のとおりである。まず、ステップ110で、クライアントProxy サーバ46のアクセスログファイル44をオープンし、書き込み位置をファイルの末尾に移動する。アクセスカウンタ変数counter=0に初期化する。AccessLog.txt という空のワークファイル56を作成する。変数MAXURL=1000 に初期化する。

【0135】続いて、ステップ112でユーザがクライアントProxy サーバ46経由でURL をアクセスすることによって発生するアクセスログよりURL 部を取り出す。そして、それらURL のうちテキストファイルオブジェクトのみワークファイルAccessLog.txt にコピーし、counter に1加算する。なおURL のうちテキストファイルオブジェクトはURL の末尾に「/」、「.htm」または「.html」等を有するものとして識別することができる。このようにテキストファイルオブジェクトのみを抽出することにより、放送用サーバ計算機22に送信されるアクセスログが少なくなり、放送サーバ計算機22で実行される処理負荷が軽減される。これについては後述する。

【0136】counter の値がMAXURL値を超えると、ステップ114でcounter をリセットし0にする。また放送サーバ計算機22で動くWWWサーバ、たとえば「http://broadcast.foo.bar.co.jp/」なるURL のサーバに対してHTTPのPOSTメソッドを使って現在のCategoryIDとユーザアクセスログが記録されたワークファイルAccessLog.txt を送り出す。このあとワークファイルAccessLog.txt を空にする。

【0137】この時の送出データ形式は次の表7に示すようになる。すなわち送出データは、現在のクライアントの使用カテゴリを示すCategoryID: ヘッダと、それに続くURL 名称とカテゴリIDとのリストである。

【0138】

【表7】

```

POST http://broadcast.foo.bar.co.jp/cgi-bin/logpost HTTP/1.0¥r¥n
¥r¥n
CategoryID: 0,5,8¥r¥n
http://www.cnn.com/                3¥r¥n
http://www.cnn.com/today/new.html   4¥r¥n
http://www.news.com/index.htm       5¥r¥n
http://www.news.com/pc/index.html   6¥r¥n
以下 URL が続く

```

【0139】放送サーバ計算機22のWWW サーバ60の仮想キャッシュヒット率算出部80は、POSTされたURL リストをもとに、放送サーバ計算機60のキャッシュ62にキャッシュされている内容と比較することで各キャッシュカテゴリごとのキャッシュヒット率を計算し、結

果をパーセント表示でクライアント計算機に返す。このときの応答データの形式の例を次の表8に示す。

【0140】

【表8】

```

-----
HTTP/1.0 200 OK\r\n
Server: NCSA/1.5\r\n
Date: Friday, 31-May-97 09:00:09 GMT\r\n
Content-type: text/plain\r\n
\r\n
Virtual Cache Hit Rate for your request\r\n
CategoryID: 1 Hitrate: 34.3%\r\n
CategoryID: 2 Hitrate: 23.4%\r\n
CategoryID: 3 Hitrate: 45.1%\r\n
CategoryID: 4 Hitrate: 31.1%\r\n
CategoryID: 5 Hitrate: 45.8%\r\n
CategoryID: 6 Hitrate: 55.1%\r\n
CategoryID: 7 Hitrate: 0%\r\n
CategoryID: 8 Hitrate: 4.1%\r\n
CategoryID: 9 Hitrate: 9.1%\r\n
-----

```

【0141】このステップにより、この例ではユーザが実際に頻繁にアクセスしているURLが属するカテゴリの上位はカテゴリIDは3、5、6であり、カテゴリIDの3、5、6がキャッシュ選別のためのカテゴリに適していることがわかる。

【0142】続いてステップ118で、ステップ116の結果に基づいてカテゴリの変更が行なわれる。初期値はCategoryID[]={0, 5, 8}となっていたが、前ステップでの仮想キャッシュヒット率の高い順から3個を選べるとCategoryID[]={3, 5, 6}となる。これにより以後は受信部ではカテゴリIDが3,5,6のもののみが選択受信されるようになる。こうしてカテゴリを変更することにより、クライアント計算機26のクライアントProxyサーバ46におけるキャッシュヒット率が高くなり、ブラウザ40を経由したアクセスが高速化できる。

【0143】なお、カテゴリIDのうち特定の番号をクライアント側で記憶しておきCategoryID変更時に選択しないようにすることも容易にできる。こうすることで、ユーザが望んでいないカテゴリの受信を回避しクライアント計算機のキャッシュ用記憶エリアを節約できる。

【0144】また、カテゴリIDのうち特定の番号をクライアント側で記憶しておきCategoryID変更時には必ずその番号が含まれるようにすることも容易にできる。こうすることにより、たとえばアクセス頻度は高くないがユーザの業務上で必須とみなせるカテゴリのファイルオブジェクトを自動選択において必ず選択するようにできる。そのため、ユーザの利用環境を最適に保つことができる。

【0145】この実施の形態ではPOSTされたCategoryID: の値は利用されていないが、後述する実施の形態2では利用する。

【0146】放送サーバ計算機22でのアクセスログ受け付けWWWサーバ60の仮想キャッシュヒット率算出部80での仮想ヒット率の計算は、以下の手順によって実

行される。なお、放送サーバでのアクセスログ受け付けWWWサーバでは、アクセスログをクライアントから受けると、CGI(Common Gateway Interface)を利用しカテゴリ制御部50からの仮想キャッシュヒットクエリに対して各カテゴリごとのキャッシュヒット率を算出して、テキスト形式でクライアントに応答する。

【0147】図5を参照して、URLカウンタ変数URLcount=0,ヒットカウント配列Hitrate[]を初期値0で始める(130)。

【0148】ユーザからPOSTされたURLリストから1行取り出し、そのURLについて、キャッシュされたファイルが存在するか否かを各キャッシュファイルについて確認する(132)。たとえば取り出されたURLが「http://www.cnn.com/index.html」であれば以下が存在するか否かを確認する。

【0149】

【表9】

```

/cache/0/http/www.cnn.com/index.html
/cache/1/http/www.cnn.com/index.html
/cache/2/http/www.cnn.com/index.html
/cache/3/http/www.cnn.com/index.html
/cache/4/http/www.cnn.com/index.html
/cache/5/http/www.cnn.com/index.html
/cache/6/http/www.cnn.com/index.html
/cache/7/http/www.cnn.com/index.html
/cache/8/http/www.cnn.com/index.html
/cache/9/http/www.cnn.com/index.html

```

この確認には、システムコールであるfopen()を用いる。取り出されたURLが存在するならHitrate[CategoryID]を1加算する。処理URL数を数えるため変数URLcountに1加算する。

【0150】ステップ134で、全てのURLに対して上述のステップ132の処理を行なったか否かを調べ、全てのURLに対しての処理が終了していなければステップ

132に戻り処理を次のURL に対して繰返す。

【0151】全てのURL に対しての処理が完了したらHitrate[]結果をURL 総数で割って%表示にする(136)。これにより、クエリを行なったクライアント計算機について、各カテゴリごとの仮想ヒット率が計算できる。この結果がクライアント計算機に返答される。

【0152】以上のようにクライアントのアクセスログをもとに放送サーバ計算機22側でカテゴリごとの仮想キャッシュヒット率を算出し、クライアントに返すことで、クライアントのカテゴリ制御部50は、最もヒット率が高くなるように、選択されるカテゴリを適応的に変更してゆくことができる。その結果、クライアント側では最小限のキャッシュサイズで最大の効果をあげることができる。

【0153】ユーザによるクライアントProxy 46経由のブラウザ利用ステップは以下になる。以下の説明はクライアントProxy 46の動作に対するものである。

【0154】ユーザ30(図1参照)がクライアント計算機26に搭載されたブラウザ40を用いて、キャッシュファイル42を有するクライアントProxy プロセス46を経由してWeb サーバ計算機24のファイルオブジェクト http://www.sharp.co.jp/をアクセスするものとする。

【0155】図6を参照して、このユーザアクセスが、クライアントProxy 46により受け付けられる(140)。

【0156】続いて、対応のファイルオブジェクトがクライアントキャッシュに存在するか否か、存在する場合にはさらにその有効期限が過ぎていないかどうかチェックされる(142)。このとき、クライアント計算機のファイルでは「http://www.sharp.co.jp/index.html」というファイルオブジェクトが「C:\%cache%\http\www.sharp.co.jp\index.html」と変換されているものとする。キャッシュの最終変更時刻はこのファイルの最終ファイル変更時刻を利用する。有効期限は予め定められているもので、変数EXPIREに保持し、例えば24時間等と設定されている。ファイルの最終ファイル変更時刻から変数EXPIREにより特定される時間が経過していればそのファイルオブジェクトは有効期限切れとする。有効期限以内のキャッシュファイルがあれば制御はステップ144に進み、なければ制御はステップ146に進む。

【0157】有効期限内のキャッシュありの場合には、ステップ144でキャッシュファイル42からオブジェクトを取り出しブラウザ40に渡す。このとき応答ヘッダ部にX-CategoryID: フィールドがあればそれは放送サーバ計算機22から放送されてキャッシュファイル42に格納されたファイルオブジェクトである。そこで、応答ヘッダからカテゴリIDを取り出し一時記憶変数Categorynow に記憶する。この一時記憶変数Categorynow

は、Web サーバ24から取得された、カテゴリIDの付されていないファイルオブジェクトに付すべきカテゴリIDとして使用される。

【0158】ファイルオブジェクトがないかまたは期限切れであればステップ146でサーバ計算機へアクセスし、最新のファイルオブジェクトを取得する。このときの手順は以下の通りである。このときには、HTTPの規約に従ってサーバ計算機のTCP/IPの80番のポートをオープンし以下のようなコマンドを書き込むとファイルオブジェクトが読み出される。

【0159】

【表10】

```
GET /index.html HTTP/1.0\r\n
```

```
\r\n
```

このオブジェクトはクライアントProxy 46のキャッシュファイル42にキャッシュされるとともにクライアント計算機26のブラウザ40に転送される。期限切れの場合の手続に関してはHTTPプロトコルに規定されている。それによれば、キャッシュファイルオブジェクトのHTTPヘッダにあるLast-Modified: フィールドの時刻を取り出してGET に追加することで最新のオブジェクトがサーバ計算機より得られる。

【0160】

【表11】例)

```
GET /index.html HTTP/1.0\r\n
```

```
If-Modified-Since: 日付\r\n
```

```
\r\n
```

この場合ファイルオブジェクトが変化していなければサーバからは以下のような応答が得られる筈である。

【0161】

【表12】

```
HTTP/1.0 304 Not modified\r\n
```

```
Date: Thu, 05 Jun 1997 06:25:53 GMT\r\n
```

```
Server: NCSA/1.5\r\n
```

```
\r\n
```

この応答が得られれば、キャッシュされているファイル「C:\%cache%\http\www.sharp.co.jp\index.html」の最終ファイル変更時刻だけを現在の時刻に更新する。

【0162】最後に、ステップ148で、クライアントProxy プロセス46は、アクセスログ44を「C:\%proxy%\log\970528.log」というファイルに記録する。ここで「970528」は1997年5月28日のログであることを意味する。アクセスログのひとつの行は以下の様になっている。

【0163】

【表13】

```
localhost - - [時刻] "GET http://www.sharp.co.jp/index.html HTTP/1.0" 200 1717 5\r\n
```

ここで、最後のフィールドの数字はX-CategoryID: フィールドのカテゴリ番号でありCategorynow 変数に記憶さ

れていたものである。

【0164】以上述べた実施の形態1では放送サーバ計算機22は、ファイルオブジェクトを複数のカテゴリに分類して蓄積し、カテゴリ識別番号とともに放送する。クライアントは、受信されたファイルオブジェクトのうち、ユーザの利用するカテゴリのものだけを選択してキャッシュに蓄積すればよい。したがって、クライアント側が小さいキャッシュ容量ですむという効果を得ることができる。たとえば各カテゴリ100MBとして100カテゴリに分類された放送サーバ計算機22側のキャッシュは10GBにもなるが、クライアント側はこれら全てを受信する必要はない。たとえば、ディスクの空きエリアを考慮して3カテゴリだけ受信することができる。この場合、クライアント側のキャッシュ容量として必要なのは300MB程度となる。しかも受信カテゴリは最大の効果を生むように自動的に変化し適応してゆくのでユーザがカテゴリを選択する操作は不要である。

【0165】またクライアント計算機は受信カテゴリIDの選択だけでファイルオブジェクトの取捨選択ができる。そのためデジタル衛星放送等高速な伝送速度による放送手段によって情報配布が行われても、クライアント側では容易にその速度に追従して取捨選択できるという利点もある。

【0166】また、カテゴリIDのうち特定の番号をクライアント側で記憶しておき、CategoryID変更時に選択しないようにしておけば、ユーザが望んでいないカテゴリの受信を回避できる。その結果、クライアント計算機のキャッシュ用記憶エリアを節約できる。

【0167】またさらに、カテゴリIDのうち特定の番号をクライアント側で記憶しておき、CategoryID変更時には必ずその番号が含まれるようにしておけば、アクセス頻度はそれ程高くないが、ユーザの業務上では必須のカテゴリが自動選択において必ず選択されるようにできる。

〔実施の形態2〕図7に、本願発明の実施の形態2にかかる分散ファイルシステムのブロック図を示す。図7において、図1に示される各構成要素と同一の構成要素には同一の参照符号を付してある。それらの機能も同一である。したがって、それらについての詳細な説明は繰返さない。

【0168】図7を参照して、この実施の形態2にかかる分散ファイルシステム160は、ネットワークに接続された複数のWebサーバ24と、放送サーバ計算機162と、複数のクライアント計算機26、28等とを含む。

【0169】放送サーバ計算機162は、実施の形態1の放送サーバ計算機22のアクセスログ受信用WWWサーバ170に代えて、クライアント計算機26、28等から受信したアクセスログに基づいて、複数のクライアント計算機に共通のURLリスト184を作成するためのUR

Lリスト蓄積部182をさらに含むアクセスログ受信用WWWサーバ170を含む。また、実施の形態1のキャッシュ送出制御部70に代えて、URLリスト184をURLの出現頻度順に並べ替えた放送リスト68を作成し、この放送リスト68により定められる順序にしたがって、キャッシュ62に蓄積されたファイルオブジェクトをクライアント計算機に向けて放送するためのキャッシュ送出制御部186を含んでいる。

【0170】この構成により、多数のクライアント計算機の望んでいる情報が優先的に放送サーバ計算機162から放送されることになる。したがって、仮想的に複数のクライアント計算機が共有するProxyキャッシュをクライアント側で実現し、ヒット率を向上させてサーバ計算機のファイルオブジェクトのアクセスを見かけ上向上させることができる。

【0171】この実施の形態によると、放送用Proxyサーバ64は個々のクライアント計算機のファイルオブジェクト取得要求を個別に受け付ける必要がない。したがって放送用サーバ計算機162がネットワーク上でクライアント計算機の近くにある必要がない。その上、一つの放送用サーバ計算機162がサポートできるクライアント計算機の上限数が大幅に増加するという効果を奏することができる。

【0172】以下、この実施の形態の分散ファイルシステムの各部の動作について具体的に説明する。クライアント計算機26からあるファイルオブジェクトへのユーザアクセスの仕方は実施の形態1とまったく同じである。したがってここでは詳細な説明は繰返さない。

【0173】放送サーバ計算機162は以下のように動作する。放送サーバ計算機162は実施の形態1の放送サーバ計算機22と同様であるが、図4のステップ114でlogpostされたアクセスログの内容から、実施の形態1で行われた仮想ヒット率を計算するだけではなく、URLリストをカテゴリ別に収集し、頻度順に並べ替える点で相違する。

【0174】アクセスログ受信用WWWサーバ170のURLリスト蓄積部182は、クライアント計算機26、28等からアクセスログの通知を監視し、URLリスト184に蓄積する。放送サーバ計算機162はこのURL情報を受け取ると放送サーバ内部のキャッシュ付きプロキシサーバ64およびネットワークを経由してWebサーバ計算機24からファイルオブジェクトを取得しカテゴリIDに応じてキャッシュファイル62にキャッシュする。カテゴリIDは、実施の形態1で述べた方法によってアクセスログに付与されている。キャッシュ送出制御部186は、URLリスト184をURLの出現頻度順に並べ替えた放送リスト68を作成し、この放送リスト68により定められる順序にしたがって、キャッシュ62に蓄積されたファイルオブジェクトをクライアント計算機に向けて放送する。

【0175】たとえば、クライアント計算機26とは別のクライアント計算機28もクライアント計算機26と同様の構成であるものとする。クライアント計算機28は、キャッシュ受信部48を持ち、ブロードキャストされたファイルオブジェクトをキャッシュ受信部48で受信しキャッシュファイル42に書き込む。このキャッシュファイル42はクライアントProxy 46のキャッシュであるので、ユーザ32サーバ計算機をアクセスするときのキャッシュとして利用しうる。すなわちクライアント計算機26と28とが仮想的にひとつのキャッシュを共有していることになるので、他のユーザのアクセス結果までキャッシュに反映できアクセスの高速化が実現できる。

【0176】すなわち本発明では、複数のクライアント計算機が利用する、共有されるキャッシュファイルを有するゲートウェイ計算機は用いられないが、ゲートウェイ計算機と同様のキャッシュ機構（すなわちファイルオブジェクトの共有機能）が個々のクライアント計算機において実現されるという効果を奏する。

【0177】以下に、放送サーバ計算機162における処理手順を述べる。頻度順リスト作成部182は、CGI

で起動されるアクセスログ受け付けプロセスlogpostとして実現される。なおURL 10000件ごとに放送部を起動するよう定数MAX=10000を持つものとする。またカテゴリ毎にURL 情報収集バッファURLlist.ID.logというファイルを持ち、行数をメモリ変数URLcounter[ID]に持つものとする。この初期値は0とする。なお、放送部の起動は、一定周期ごとにするにしてもよく、また蓄積URLのうち、異なるURL が一定数となったときにおこなっても、または手動で行なってもよい。

【0178】図8を参照して、アクセスログ受け付けプロセスのステップ190で、アクセスログ受信用WWWサーバ（http://broadcast.foo.bar.co.jp/cgi-bin/logpost）の仮想キャッシュヒット率算出部80は、POSTされたURL リストをもとに各キャッシュカテゴリでのキャッシュヒット率を計算し、結果をパーセント表示でクライアント計算機に返す。これは実施の形態1と同じである。この計算の結果、カテゴリ別の仮想キャッシュヒット率が次のように求められるものとする。

【0179】

【表14】

```

.....
HTTP/1.0 200 OK¥r¥n
Server: NCSA/1.5¥r¥n
Date: Friday, 31-May-97 09:00:09 GMT¥r¥n
Content-type: text/plain¥r¥n
¥r¥n
Virtual Cache Hit Rate for your request¥r¥n
CategoryID: 1 Hitrate: 34.3%¥r¥n
CategoryID: 2 Hitrate: 23.4%¥r¥n
CategoryID: 3 Hitrate: 45.1%¥r¥n
CategoryID: 4 Hitrate: 31.1%¥r¥n
CategoryID: 5 Hitrate: 45.8%¥r¥n
CategoryID: 6 Hitrate: 55.1%¥r¥n
CategoryID: 7 Hitrate: 0%¥r¥n
CategoryID: 8 Hitrate: 4.1%¥r¥n
CategoryID: 9 Hitrate: 9.1%¥r¥n
.....

```

【0180】このステップにより、実施の形態1の場合と同様に、ユーザが実際に頻繁にアクセスしているURL はカテゴリIDが3, 5, 6のものであること、したがってカテゴリIDの3, 5, 6が、クライアント計算機に設定するものとして適していることがわかる。

【0181】続いて、ステップ192で、POSTメソッドでクライアントから送られてきたURL リストを仕分けしカテゴリ別にURLlist.ID.logファイル186に追加する。ここで「ID」はカテゴリIDであり、この実施の形態では0から9までである。また変数URLcounter[ID] (ID=3,5,6) に、送られてきた各カテゴリの行数分のURL 数を加算する。この処理が、図7に示す頻度順リスト作成部182の処理のうち、URL の蓄積処理に相当する。

以上でlogpost の動作は終了する。

【0182】次にキャッシュ送出制御部186は以下のように動作する。なお、キャッシュ送出制御部186は、各カテゴリごとに起動されたプロセスにより実現されるものとする。ここではカテゴリ2用のキャッシュ送出制御部186の動きを例として説明する。

【0183】図9を参照して、URLcounter[2] > MAXかどうかを判定する(210)。判定結果がYESであれば十分多くのURL 情報が蓄積されたものと考えられるので、サーバ計算機24へのアクセスと、キャッシュされたファイルオブジェクトの放送とを開始するためステップ214以下の処理を行う。判定結果がNOであればsleep 60秒として休止し(212)ステップ21

0に制御を戻す。

【0184】ステップ214では頻度順処理が行われる。URL リストであるURLlist.2.log184には次のよ

うなアクセスログが蓄積されている。

【0185】

【表15】

```
http://naragw.sharp.co.jp/sps/index.html¥r¥n
http://www.sharp.co.jp/¥r¥n
http://www.cnn.com/today/new.html¥r¥n
http://www.news.com/index.htm¥r¥n
http://www.news.com/pc/index.html¥r¥n
以下出現頻度順に URL が続く
```

【0186】ステップ214では、このアクセスログをURL の出現頻度順に並べ替える。本実施の形態では、放送サーバ計算機162としてUNIX OSを用いたものが想定されているので、sort, uniq, awk 等のUNIX標準装備

```
cat URLlist.2.log | sort | uniq -c | sort -n -r | awk ' {print $2} ' > broadcast.2.log ... (7)
```

このコマンドによって、アクセスログをURL の出現頻度順に並べ替えることができる。こうして得られたファイルを放送リスト(broadcast.2.log)68として、キャッシュファイル62内のファイルオブジェクトをキャッシュ送出制御部186から送出する。この処理により、放送するオブジェクトはユーザのアクセス頻度の高いオブジェクトから優先的に放送されることになる。

【0188】なお、上記処理にパターンマッチングのフィルタを入れてある特定のパターンにマッチするものは放送リストbroadcast.2.log から排除することもできる。たとえば排除文字列がファイル/reject.txt 200

```
cat URLlist.2.log | fgrep -v /reject.txt | sort | uniq -c | sort -n -r | awk ' {print $2} ' > broadcast.2.log ... (8)
```

続いてステップ216でサーバアクセスを行う。具体的には、放送リスト68であるbroadcast.2.log をオープンし、放送リスト68に記載されているURL にしたがった順次、放送用Proxy サーバhttp://broadcast.foo.bar.co.jp:10082/経由でWeb サーバ計算機をアクセスする。なおProxy サーバのキャッシュのうち、/cache/2/ というディレクトリ内にカテゴリ2のためのキャッシュファイルを作成する。「/cache/2/」の「2」はカテゴリ番号を意味する。

【0191】すなわちこのステップでは、サーバbroadcast.foo.bar.co.jp のポート10082に対しコネクションをはって以下を書き込む。

【0192】

【表17】

```
GET http://naragw.sharp.co.jp/sps/index.html HTTP/1.0¥r¥n
¥r¥n
```

これにより放送用Proxy サーバ64がWeb サーバ計算機

のコマンドを使ってこの処理を実現できる。たとえば以下のコマンドをOSに対して発行すればよい。

【0187】

【数7】

```
| sort -n -r | awk ' {print $2}
```

にあり、その内容が以下の通りであるとする。

【0189】

【表16】

```
www.xxxx.yyyy.com
.adult
porn
```

この排除リストにより、fgrep コマンドを使って特定のパターンを含むURL を放送リストから除去することができる。コマンドは以下のようになる。

【0190】

【数8】

から放送リストのURLで特定されるファイルオブジェクトを取得しキャッシュする。また、応答データとしてファイルオブジェクトが読み出せる。

【0193】次に、ステップ218で、カテゴリIDとURL 名ヘッダとを付加してこのファイルオブジェクトを放送する。カテゴリIDはX-CategoryID:、URL ヘッダはX-URL:というヘッダを識別子として使用する。URL ヘッダはキャッシュ放送においてどのURL オブジェクトかを識別するために使用するもので、キャッシュファイル名からURL を復元する。これは実施の形態1の場合と同様である。

【0194】たとえば、キャッシュされたファイルオブジェクトが「/cache/2/http/naragw.sharp.co.jp/sps/index.html」の場合には、放送されるオブジェクトの内容は以下のようになる。

【0195】

【表18】

```

-----
HTTP/1.0 200 OK\r\n
X-CategoryID: 2\r\n
X-URL: http://www.naragw.sharp.co.jp/sps/index.html\r\n
Server: NCSA/1.5\r\n
Date: Friday, 30-May-97 07:00:09 GMT\r\n
Last-Modified: Friday, 30-May-97 01:03:52 GMT\r\n
Content-Length: 49763\r\n
Content-type: text/html\r\n
\r\n
<html>\r\n
<head><title>Super Proxy Script</title></head>\r\n
<body>\r\n
<h1>Super Proxy Script</h1>\r\n

```

【0196】以下、HTMLのテキストデータが続く。このデータが放送ハードウェア部72に送出される。また、累積放送バイト数を加算する。

【0197】最後に、放送リスト68であるbroadcast.2.logの末尾に達したか、または規定累積バイト数を越えたか、または規定時間を越えたか、が判定され、いずれの条件も成立していなければステップ216に制御を戻し、ステップ216以下の処理を繰返す。以上が本実施の形態の分散ファイルシステムの動作である。

【0198】この実施の形態2では、放送されるキャッシュファイルオブジェクトはユーザのアクセスログから収集したもので、かつ各カテゴリに関連付けされた情報である。そのためあるカテゴリを選んでいるユーザには他のユーザのアクセスしたものも含めたそのカテゴリの情報のみが伝達される。従ってユーザの選択しているカテゴリのURL情報を、他人のアクセス結果まで反映してキャッシュすることができる。また実施の形態1と同様にキャッシュヒット率が高くなるように、クライアント計算機で設定されるカテゴリが、アクセス結果を反映して自動的に変更されるので、ユーザの嗜好にしたがって自然に設定カテゴリを変えてゆくことが可能となる。

【0199】また放送内容を決定する場合に頻度順処理やフィルタ処理をかけることで有限の放送時間内に最も効果的な放送内容から先に放送できるので、クライアント計算機におけるキャッシュ効果を高くすることができるというメリットがある。

〔実施の形態3〕本願発明の実施の形態3にかかる分散ファイルシステムの概略構成を図10に示す。この実施の形態3の分散ファイルシステム240では、実施の形態2のキャッシュ放送手順の中でサーバ計算機をアクセスして最新情報に更新した際に、ファイルオブジェクトの最終変更日付が得られるのでこれを別途放送する。そして、クライアントのキャッシュをこの放送された最終変更日付を用いて適宜変更したり、ファイルオブジェクトを取得したりして最新状態に保つ点に特徴がある。

【0200】一般に、放送用Proxyサーバのアクセスログではサーバ計算機の状態コードが記録されている。そして、状態コードが304番の場合、ファイルオブジェクトに変化がなかったということを意味する。そこで、放送用ProxyサーバのキャッシュのファイルオブジェクトのHTTPヘッダ部からLast-Modified: フィールドにある日付(Ltime)を抽出し、これをこのファイルオブジェクトの最終変更日付として放送する。日付の形式は「Friday, 30-May-97 01:03:52 GMT」等とし、Last-Modified: フィールドと同じものとする。データ形式は通常のHTTPと同じである。またヘッダとして「VALID」を使用する。このようにヘッダに「VALID」を使用するので、この最終変更日付の放送を仮に「VALID 放送」と名付ける。

【0201】VALID 放送データの例を以下の表に示す。

【0202】

【表19】

```

HTTP/1.0 200 OK\r\n
X-URL: http://broadcast.foo.bar.co.jp/cache/\r\n
Content-type: text/html
\r\n
VALID http://www.sharp.co.jp/image/mebius/index.html Friday, 30-May-9701:03:52 GMT\r\n
VALID http://www.sharp.co.jp/image/zaurus/index.html Friday, 30-May-97 01:03:52 GMT\r\n
VALID http://www.foo.bar.co.jp/sample/aaa.gif Friday, 30-May-97 01:03:52 GMT\r\n
VALID http://www.foo.bar.com/news/header.gif Friday, 30-May-97 01:03:52 GMT\r\n
VALID http://www.news.foo.com/weather/japan/index.html Friday,30-May-9701:03:52 GMT\r\n

```

図10を参照して、この分散ファイルシステム240は、Webサーバ計算機24と、放送サーバ計算機242と、複数のクライアント計算機246、248等を含む。

【0203】放送サーバ計算機242の構成は、図7に示す実施の形態2の放送サーバ計算機162とほぼ同様であるが、放送時に得られたファイルオブジェクトの最終変更時刻を放送する機能をさらに有するキャッシュ送出制御部250をキャッシュ送出制御部186に代えて有する点で異なっている。またクライアント計算機246の構成は、図7に示すクライアント計算機26とほぼ同様であるが、放送サーバ計算機242から放送される、ファイルオブジェクトの最終変更時刻の情報（VALID 放送）を受信してこの最終変更時刻の情報のみを他と区別して抽出する機能を有するキャッシュ受信部252をキャッシュ受信部48に代えて含む点、このキャッシュ受信部252が、最終変更時刻の情報にしたがって、キャッシュファイル42の該当ファイルの最終ファイル変更時刻を更新したり、クライアントProxy 46を経由してファイルオブジェクトをWeb サーバ24から取得してキャッシュさせたりする機能をさらに有する点、で異なっている。

【0204】以下、クライアント計算機246におけるキャッシュ受信部252の動作を図11とともに説明する。前述したとおり、キャッシュ受信部252は、VALID 放送を受信した場合、キャッシュファイルに書き込むのではなく、これはVALID 放送であると解釈して以下の動作を行なう。

【0205】まずステップ260で、受信したVALID 放送から作成したリスト（VALID リストと呼ぶ）の1行を読む。

【0206】読出した情報に基づいてクライアントキャッシュファイル42をアクセスし、キャッシュされたファイルがあればそのLast-Modified: フィールドの時刻とLtime を比較する。両者が同じならばキャッシュされているファイルは有効であると判断し、さもなければ無効であると判断する（262）。有効のときは制御はステップ266に、無効のとき制御はステップ264に、それぞれ進む。

【0207】有効と判断されたときは、ステップ266でこのキャッシュされているファイルの最終ファイル変更時刻を現在の値に更新する。この処理の結果、現在から有効期限以内の時間にわたって再びキャッシュが有効となる。すなわち、現時点でこのファイルオブジェクトをWeb サーバから取寄せた場合と同じ効果を生ずる。

【0208】例えばクライアントキャッシュされているファイルの最終ファイル変更時刻が「1997年6月20日20時30分10秒」であれば、このキャッシュされているファイルのパスおよびファイル名を用いてクライアントOSのシステムコールを呼んでこのファイルの最終ファイル変更時刻を現在時刻、たとえば「1997年6月21日21時12分30秒」に変更する。この処理により、あたかも1997年6月21日21時12分30秒にこのクライアントがこのファイルオブジェクトをWeb サーバから取得したかのような効果

を生ずる。すなわち、仮にキャッシュ有効期限を示す変数EXPIRE=24時間であれば、キャッシュ有効期限が「現在の時刻1997年6月21日21時12分30秒+ 24時間」として計算しなおされる。この結果、1997年6月21日21時12分30秒までキャッシュファイルオブジェクトが有効に変更されたことになる。もちろん、放送用プロキシサーバでWeb サーバをアクセスして最新性を確認した時刻が本当に正確な最新性確認時刻であるので、確認時刻と放送時刻とが数時間以上開くようならば最新性確認時刻もURLとともに放送してもよい。

【0209】また無効と判断されるときは、キャッシュされているファイルオブジェクトが古い可能性がある。そこで、ステップ264でサーバ計算機をアクセスしてファイルオブジェクトを取得し最新情報に更新する。

【0210】ステップ262、264、および266の処理によって、ユーザの介在なしに能動的にキャッシュを更新することができる。

【0211】ただし、常時ネットワーク接続されていないクライアント計算機ではステップ264を省略しても良い。その代わりキャッシュされているファイルオブジェクトの日付を現在時刻からキャッシュ有効期限を減算した時間とし、強制的にそのキャッシュファイルオブジェクトを期限切れとする。こうすると、次回にユーザがそのファイルオブジェクトをアクセスしたときにキャッシュされているファイルオブジェクトの期限の評価が行われ、ステップ264の処理と同じ処理が行われる。

【0212】筆者らの観測によればインターネット上で変化するファイルオブジェクトの割合は1日あたり5%程度である。つまり、放送された最終変更時刻情報のうち5%程度がこのステップ264の処理を引き起こすだけである。従ってこれらのファイルオブジェクトに対してユーザアクセス時にキャッシュファイルの期限確認をおこなっても5%程度に過ぎず、影響は小さい。

【0213】再び図11を参照して、ステップ268でVALID リストの最後まで処理が終了したか否かを判定し、最後まで終了していなければステップ260以下の処理を繰返す。こうして、受信されたVALID リストの全ての行に対して上記操作を行う。

【0214】以上のような操作を行えば、放送サーバで情報の更新確認処理がされたファイルオブジェクトの状態がクライアント計算機のキャッシュにも反映される。インターネット上のWWW サーバのオブジェクトは一日あたり5%程度しか変化しないので、95%程度のオブジェクトはVALID 放送を受信するだけで自動的に最新状態であることが保証される。

【0215】低速ネットワーク（例えば転送速度128kbpsのネットワーク）を経由した場合、ファイルの最新性確認には平均5秒を要することが観測されている。一方、有効期限以内のファイルオブジェクトに対するアクセスがあったとき、クライアントProxy を経由している

と、ネットワークアクセスが発生しないので0.1秒程度でキャッシュファイルからブラウザへ当該ファイルオブジェクトを取出しうる。すなわち、本実施の形態によってキャッシュされている大部分のファイルオブジェクトについて最新性が保証されていると、サーバ計算機に対する問い合わせが抑制され応答速度が向上する。本実施の形態では、VALID 放送によってキャッシュの95%が最新状態であると保証されれば、キャッシュヒット率50%のときURLあたりのアクセス時間短縮効果は以下のように算出できる。

【0216】

【数9】

$$0.5 \times 0.95 \times 5 \text{ 秒} = 2.5 \text{ 秒} \quad \dots (9)$$

また放送を使っているのでクライアント計算機の数が多い数であってもキャッシュ更新処理のトラフィックの増加が少ないという利点もある。VALID 放送自体は情報量がURL あたりURL 長さ+時刻情報バイト数である。1 URL あたりのデータ長が100バイトとすると、総バイト数は以下の式で計算できる。

【0217】

【数10】

$$100 \text{ バイト} \times \text{URL 数} \quad \dots (10)$$

したがって100万URL（キャッシュ蓄積量10GBに相当する）分のVALID 放送をしても、その情報量はわずかに100Mバイトであり、1Mbps 程度の速度で放送すれば1000秒程度で放送終了する。したがって使用する放送帯域がわずかですみ、コストが低くすむという効果も奏することができる。

【図面の簡単な説明】

【図1】本発明の実施の形態1に係る分散ファイルシステムの構成の概略をブロック図形式で示す図である。

【図2】実施の形態1の放送サーバ装置で行なわれるファイルオブジェクトの放送処理のフローチャートである。

【図3】実施の形態1のクライアント装置で行なわれるキャッシュ受信処理のフローチャートである。

【図4】実施の形態1のクライアント装置で行なわれるアクセスログ送信処理のフローチャートである。

【図5】実施の形態1の放送サーバ装置で行なわれる仮想キャッシュヒット率算出処理のフローチャートであ

る。

【図6】実施の形態1のクライアント装置でのクライアントProxy によるユーザアクセスの処理の流れを示すフローチャートである。

【図7】実施の形態2に係る分散ファイルシステムの構成の概略をブロック図形式で示す図である。

【図8】実施の形態2のアクセスログ受付処理のフローチャートである。

【図9】実施の形態2の放送サーバ装置で行なわれる放送処理のフローチャートである。

【図10】実施の形態3に係る分散ファイルシステムの構成の概略をブロック図形式で示す図である。

【図11】実施の形態3のクライアント装置におけるVALID 放送受信処理のフローチャートである。

【図12】複数クライアントによるキャッシュ共有を採用した従来の分散ファイルシステムをブロック図形式で示す図である。

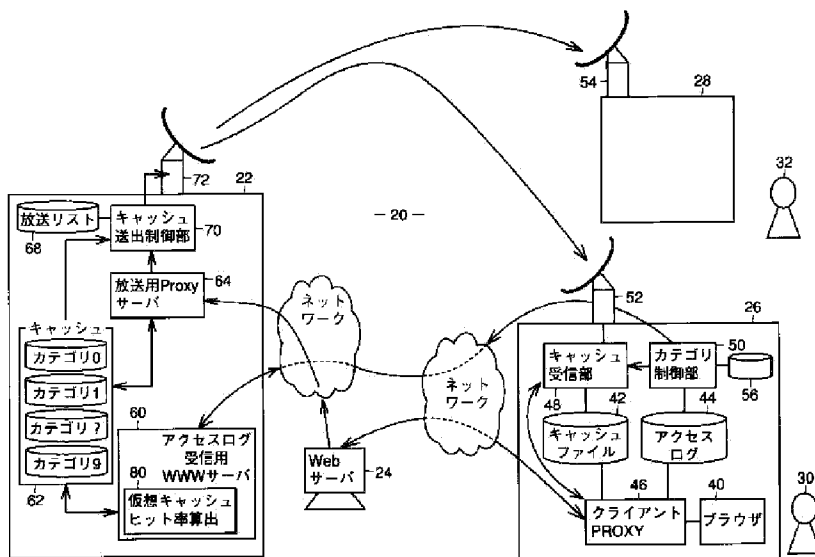
【図13】一般的なProxy サーバのハードウェア構成を示す図である。

【図14】クライアント装置にProxy サーバを内蔵した場合の分散ファイルシステムをブロック図形式で示す図である。

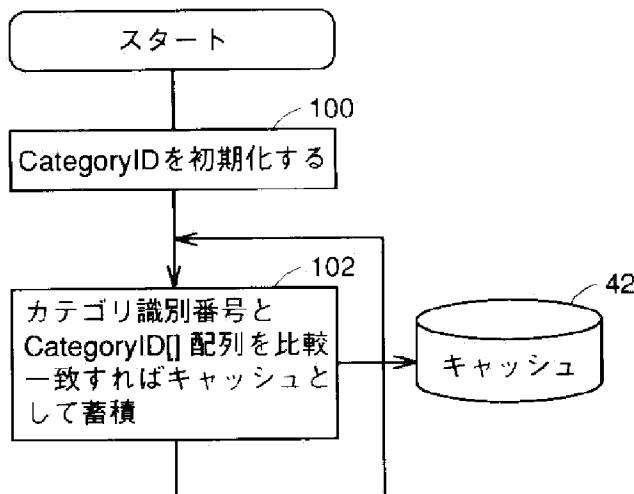
【符号の説明】

- 20、160、240 分散ファイルシステム
- 22、162、242 放送サーバ計算機
- 26、28、246、248 クライアント計算機
- 42 キャッシュファイル
- 44 アクセスログ
- 46 クライアントProxy
- 48、252 キャッシュ受信部
- 50 カテゴリ制御部
- 62 キャッシュファイル
- 64 放送用Proxy サーバ
- 68 放送リスト
- 60、170 アクセスログ受信用WWW サーバ
- 70、286、250 キャッシュ送出制御部
- 80 キャッシュヒット率算出部
- 182 URL リスト蓄積部
- 184 URL リスト

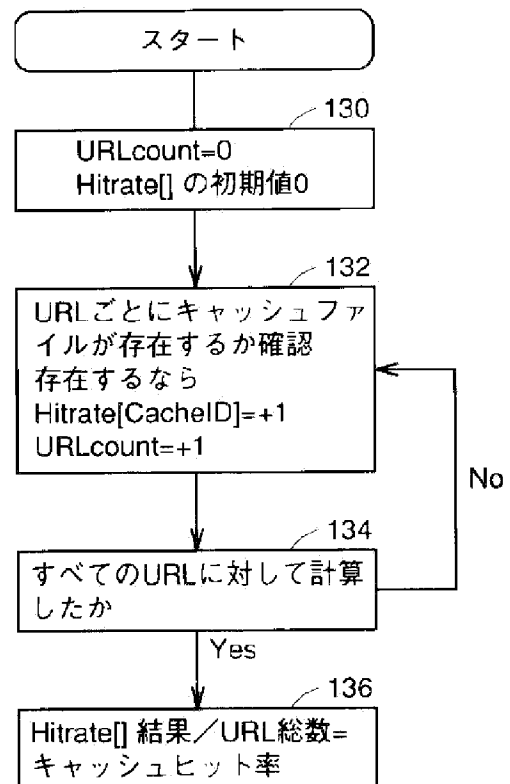
【図1】



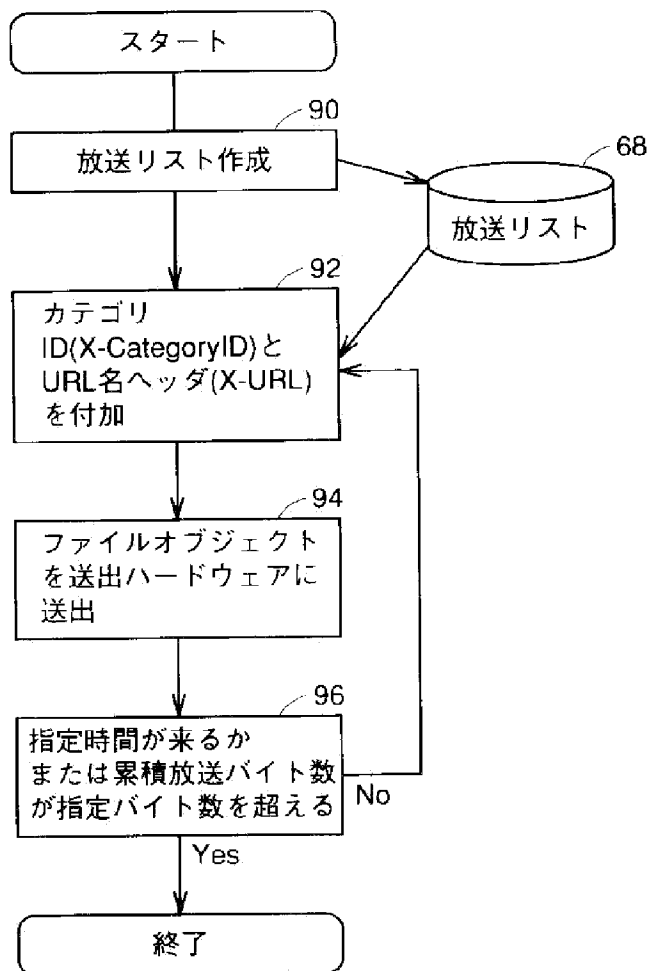
【図3】



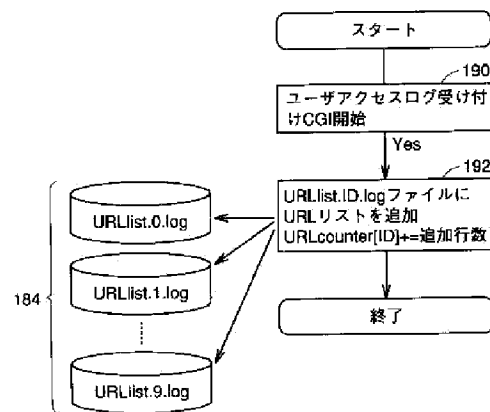
【図5】



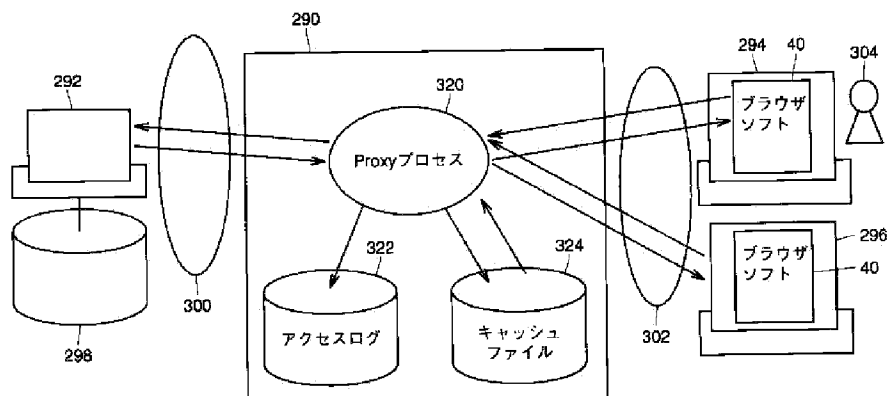
【図2】



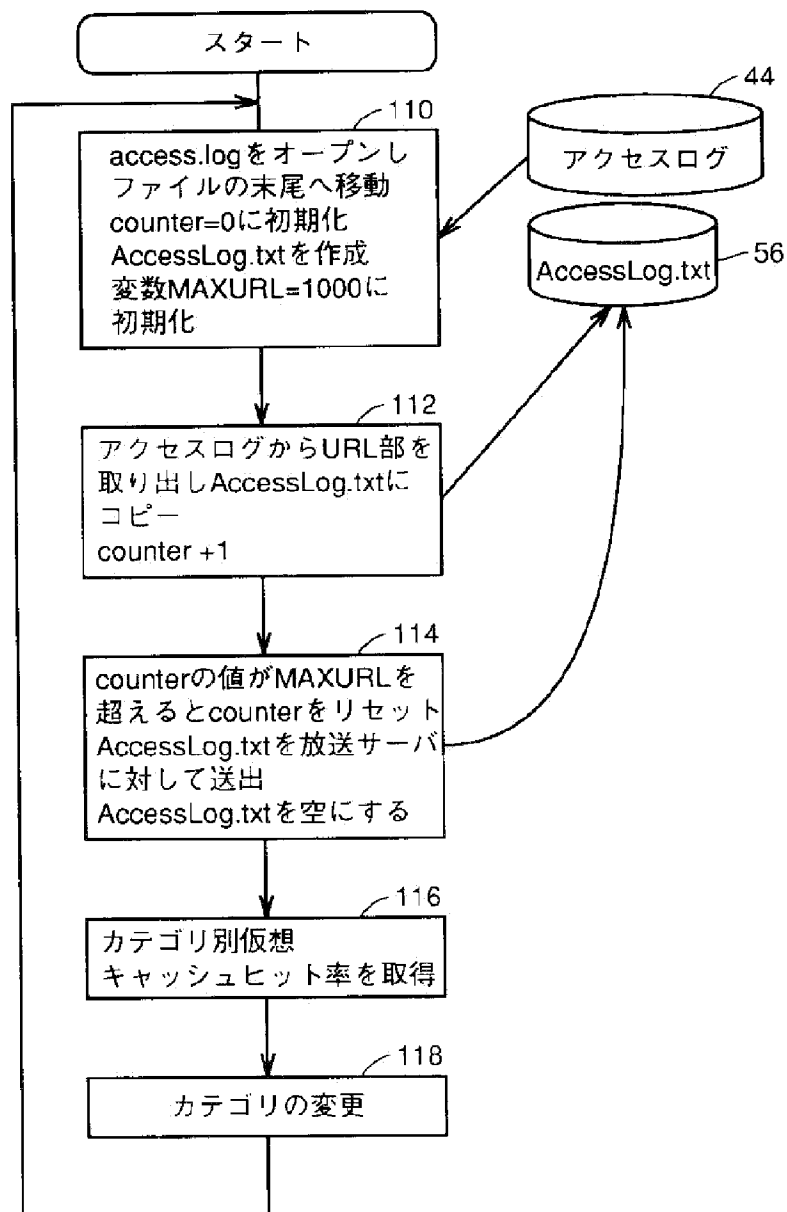
【図8】



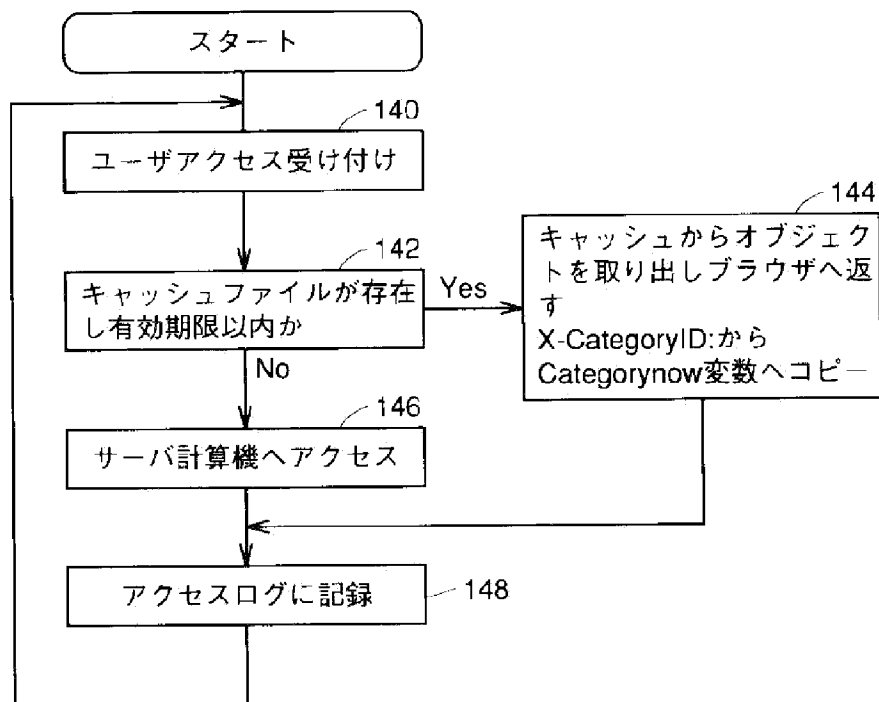
【図12】



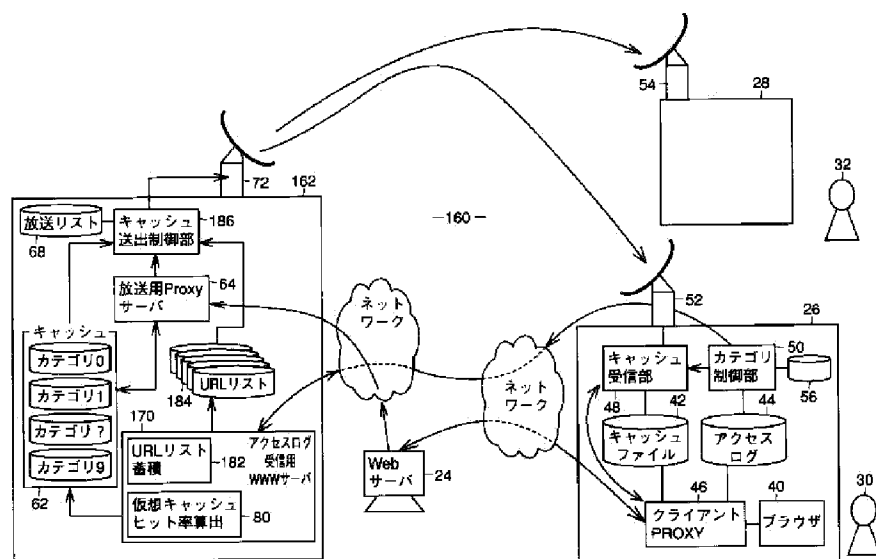
【図4】



【図6】



【図7】

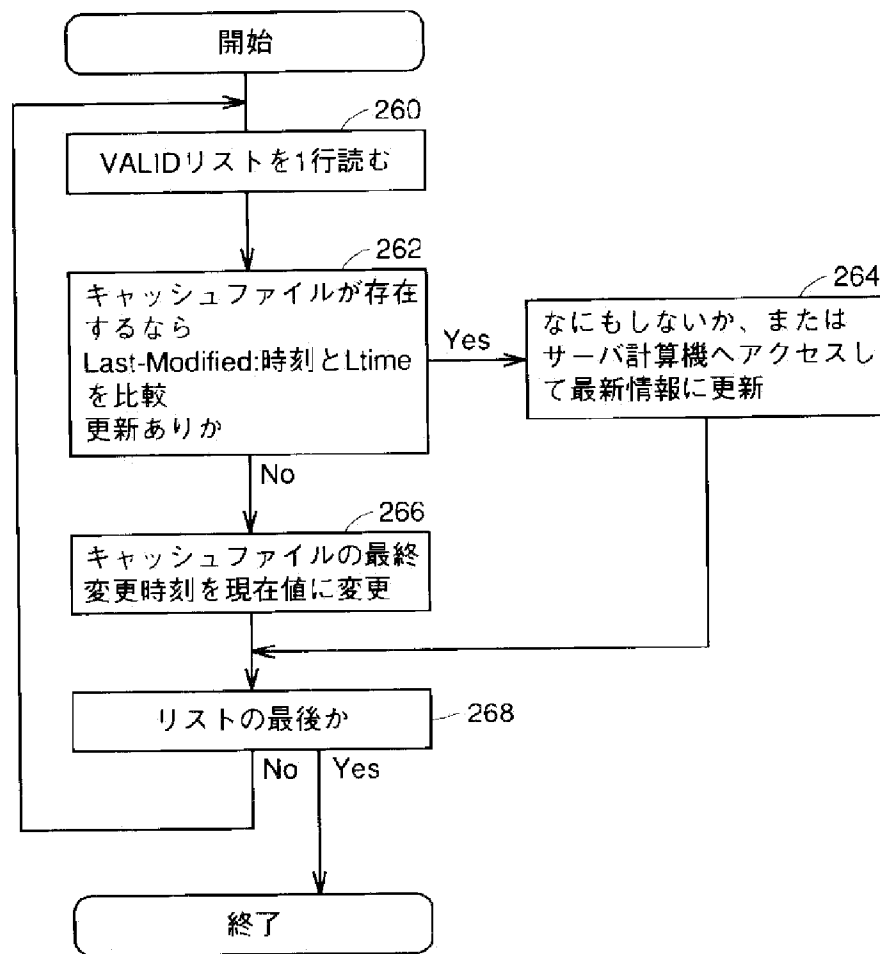



```

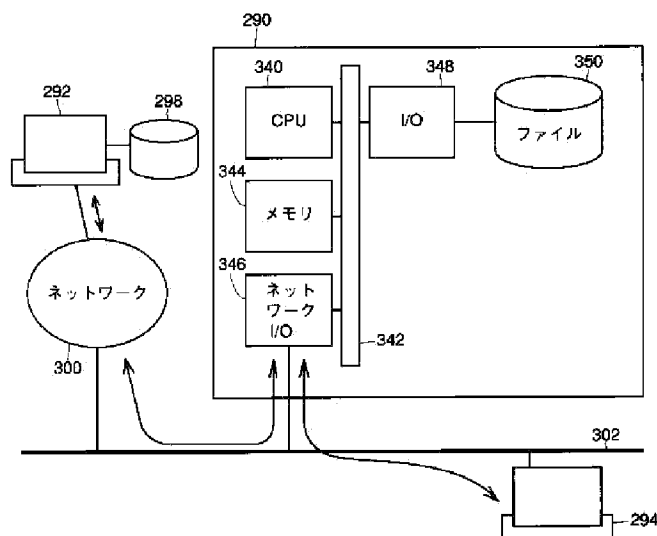
graph TD
    Start([スタート]) --> 210{URLcounter[ID]>MAXか}
    210 -- Yes --> 212[sleep 60秒]
    212 --> 210
    210 -- No --> 214[頻度順処理およびパターンマッチングによる排除処理]
    
    subgraph 184 [ ]
        200[(排除パターンファイル reject.txt)]
        U0[(URLlist.0.log)]
        U1[(URLlist.1.log)]
        U9[(URLlist.9.log)]
    end
    184 --> 214
    
    214 --> subgraph 68 [ ]
        B0[(broadcast.0.log)]
        B1[(broadcast.1.log)]
        B9[(broadcast.9.log)]
    end
    
    214 --> 216[サーバアクセス]
    216 --> 218[放送処理]
    218 --> 220{累積バイト数が規定を超えたか  
またはbroadcast.ID.logの末尾に達したか}
    220 -- Yes --> End([終了])
    220 -- No --> 214
  
```

-240-

【図11】



【図13】



【図14】

